3-23-2009

Uncooled Infrared Photon Detection Concepts and Devices

Viraj Vishwakantha Jayaweera Piyankarage

UNCOOLED INFRARED PHOTON DETECTION CONCEPTS AND DEVICES

by

VIRAJ VISHWAKANTHA JAYAWEERA PIYANKARAGE

Under the Direction of Unil Perera

ABSTRACT

This work describes infrared (IR) photon detector techniques based on novel semiconductor device concepts and detector designs. The aim of the investigation was to examine alternative IR detection concepts with a view to resolve some of the issues of existing IR detectors such as operating temperature and response range. Systems were fabricated to demonstrate the following IR detection concepts and determine detector parameters: (i) Near-infrared (NIR) detection based on dye-sensitization of nanostructured semiconductors, (ii) Displacement currents in semiconductor quantum dots (QDs) embedded dielectric media, (iii) Split-off band transitions in GaAs/AlGaAs heterojunction interfacial workfunction internal photoemission (HEIWIP) detectors. A far-infrared detector based on GaSb homojunction interfacial workfunction internal photoemission (HIWIP) structure is also discussed. Device concepts, detector structures, and experimental results discussed in the text are summarized below.

Dye-sensitized (DS) detector structures consisting of n-TiO_2/Dye/p-CuSCN heterostructures with several IR-sensitive dyes showed response peaks at 808, 812, 858, 866, 876, and 1056 nm at room temperature. The peak specific-detectivity (D^*) was 9.5×10^{10} cm $Hz^{-1/2}$ W^{-1} at 812 nm at room temperature.

Radiation induced carrier generation alters the electronic polarizability of QDs provided the quenching of excitation is suppressed by separation of the QDs. A device constructed to illustrate this concept by embedding PbS QDs in paraffin wax showed a peak D^* of 3×10^8 cm $Hz^{1/2}$ W^{-1} at ~540 nm at ambient temperature.

A typical HEIWIP/HIWIP detector structures consist of single (or multiple) period(s) of doped emitter(s) and undoped barrier(s) which are sandwiched between two highly doped contact layers. A *p*-GaAs/AlGaAs HEIWIP structure showed enhanced absorption in NIR range due to heavy/light-hole band to split-off band transitions and leading to the development of GaAs based uncooled sensors for IR detection in the 2-5 μm wavelength range with a peak D^* of 6.8×10^5 cm $Hz^{1/2}$ W^{-1}.

A HIWIP detector based on *p*-GaSb/GaSb showed a free carrier response threshold wavelength at 97 μm (~3 THz)with a peak D^* of 5.7×10^{11} cm $Hz^{1/2}$ W^{-1} at 36 μm and 4.9 K. In this detector, a bolometric type response in the 97 - 200 μm (3-1.5 THz) range was also observed.

INDEX WORDS: Infrared detectors, Photon detection, NIR detectors, THz detectors, Uncooled detectors, Dye-sensitized, IR dye, Quantum dot, Split-off band, GaSb, GaAs, AlGaAs, TiO_2, CuSCN, PbS, Homojunction, Heterojunction, Workfunction, Photoemission, Displacement currents, 1/f noise.

UNCOOLED INFRARED PHOTON DETECTION CONCEPTS AND DEVICES

by

VIRAJ VISHWAKANTHA JAYAWEERA PIYANKARAGE

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2009

UNCOOLED INFRARED PHOTON DETECTION CONCEPTS AND DEVICES

by

VIRAJ VISHWAKANTHA JAYAWEERA PIYANKARAGE

Major Professor: Unil Perera

Committee: Vadym M. Apalkov
Douglas Gies
Xiaochun He
Kirthi Tennakone
Brian D. Thoms

Electronic Version Approved:

Office of Graduate Studies
College of Arts and Science
Georgia State University
May 2009

DEDICATION

To my parents and teachers

ACKNOWLEDGEMENTS

I would like to express my deepest sense of gratitude to my supervisor Prof. Unil Perera for his patient guidance, encouragement and excellent advice throughout this study. His constant enthusiasm, support, and invaluable suggestions made this work successful. He has been everything that one could want in an advisor. Next, I am grateful to my co-supervisor, Prof. Kirthi Tennakone, who introduced me to the research field and has been abundantly helpful and offered invaluable assistance, support, and guidance. Deepest gratitude is also due to the members of the supervisory committee: Dr. Vadym M. Apalkov, Dr. Douglas Gies, Dr. Xiaochun He, and Dr. Brian D. Thoms for fruitful discussions, technical assistance, and effort in reading and providing me with valuable comments on this dissertation. I would like to convey special acknowledgement to Dr. Steven Matsik, who was always open for questions and discussions relating theoretical work and his valuable scientific advice. Special thanks to post doctoral associate Dr. Gamini Ariayawansa, former members Dr. Mohamed Rinzan, and Dr. Aruna Weerasekara for valuable discussions on the subject matters. I would also like to thank present colleagues in the group: Ranga Jayasinghe, Dulipa Pitigala, Laura Byrum, Greggory Rothmeier, Jiafeng Shao, and Dr. Manmohan Singh. I wish to thank all the individuals of the Department of Physics and Astronomy, especially the Physics workshop team headed by Mr. Charles Hopper who converted my mechanical drawings into physical reality, and friends for their contribution, friendship, love, and helps. Finally, I take this opportunity to express my profound gratitude to my beloved parents, parent in law, and my wife for their moral support and patience during my study at GSU.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BLIP background kimited performance

BPR bromopyrogallol red

CB conduction band

CQD colloidal quantum dot

CTO conducting tin oxide

CUSCN Copper thiosinate

DS dye-sensitized

DSN dye-sensitization

DSNIRPDs Dye-sensitized near infrared photon detectors

DSSCs dye-sensitized solar cells

eV electron volts

FDN frequency dependent noise

FFT Fast Fourier Transform

FIR far-infrared

FT-IR Fourier-transform infrared

GDSSCs Gratzel type solar cells

HEIWIP heterojunction interfacial workfunction internal photoemission

H-H heavy hole

HIWIP homojunction interfacial workfunction internal photoemission

IR Infrared

L-H light hole

MC mercurochrome

N3 cis-dithiocyanatobis (4-4'-dicarboxylic acid-2, 2'Ru[II])

NEP Noise Equivalent Power

NIR near-infrared

QDs Quantum dots

RH relative humidity

TEC thermo-electric-coolers

THz Terahertz

TiO_2 Titanium dioxide

UV ultraviolet

VB valence band

Chapter 1

Introduction

Humans and most animals have five senses: sight, hearing, taste, smell, and touch as avenues for observing the environment. Nature has given us one of the world's best visible cameras (equivalent to ~576 megapixels[1,2]), our eyes. Our skin is sensitive to infrared (IR) radiation. Some animals such as rattlesnakes[3] and pit vipers[4] are specialized in IR sensing. Their IR sensory organs can readily identify warm-blooded prey they need to hunt quickly. Nature invented these precision IR detection techniques millions of years ago; man noticed the presence of an invisible heat. In 1800, Sir Frederick William detected IR radiation using a thermometer[5,6]. Man-made contraptions have enabled expanding the region of detectability of the electromagnetic radiation beyond the short and long wavelength limits. In IR detectors, the radiation incident on a probe is converted to a signal readily accessible to the human senses. As an example, night vision IR imaging cameras assign visible colors that are proportional to the temperature of the object (or IR energy received by each tiny mesa (pixels) on the detector array). Eventually a night vision camera produces an image on a screen that the human brain can identify based on eyesight.

The IR region is one of the most interesting bands in the electromagnetic spectrum because all objects above a temperature of 0 K emit radiation falling within the IR range. Although a wide variety of techniques have been used to detect IR radiation, all these IR detectors can be categorized as thermal or photon devices. Thermal detectors utilize temperature-dependent properties of the detector material[7], such as thermocouples and thermopiles which use the thermoelectric effect. Bolometers and microbolometers[8] are based on changes in resistance, Golay cells[9,10] follow thermal expansion, and pyroelectric detectors[11] utilize the temperature dependent spontaneous electric polarization that generates electric charges upon incidence IR irradiation (known as pyroelectric effect).

Photon detectors have much higher sensitivity and faster response-time compared to thermal detectors[7]. These utilize electronic excitations in a semiconductor upon IR illumination as the basis of the detection method. The excited carriers can be measured as an electric potential or current change in the external circuit. Photon detectors can be divided into two types: photovoltaic and photoconductive. Photovoltaic-type detectors generate a photocurrent upon illumination, and they are usually constitute of *p*-*n* or *p*-i-*n* type semiconductor junctions, where *p*, i, and *n* denotes the *p*-doped, intrinsic, and *n*-doped semiconductors, respectively. In photoconductive detectors, changes in the conductivity (or resistivity) are monitored upon illumination. Typically, this requires an external bias voltage and a load resistor to convert conductivity changes into voltage changes which are easily measurable.

Initial photoconductive IR detectors were developed utilizing the band-to-band transitions or dopant-to-band transitions. The band-to-band detectors typically require semiconductors whose band edge is close to the energy of the photon to be detected. For sensing the IR region of the spectrum, such detectors are typically fabricated of materials, which have difficulties associated with material quality. HgCdTe and InSb are interesting materials which have been extensively studied[12] as photoconductive as well as photovoltaic systems. Cryogenically cooled HgCdTe has a broad response range from 1 μm to 20 μm and InSb responds up to 5.5 μm at the same temperature. Research is in progress to achieve uncooled HgCdTe detectors[13]. Extended InGaAs *p*-i-*n* photodiodes responding up to 2.6 μm are commercially available. On the other hand, dopant-to-band detectors operate at very low temperatures. In order to overcome the limitations of those two methods intersub-band transition based detectors[14,15] were introduced about two decades ago. Quantum well detectors utilize intersub-band transitions, but are restricted to the absorption of normal incident radiation due to the selection rules[16]. Hence, these detectors require 45° coupling or a corrugated surface (for grating coupling). Furthermore, quantum efficiency is greatly reduced due to the selection rule permitting only one polarization mode of the incident light. Subsequently homo[17] and hetero[18] junction interfacial workfunction internal photoemission (HIWIP/ HEIWIP) detectors were introduced. These utilize intra-band transitions (light-hole to heavy-hole transitions) and are free of selection rule limitations. Since the free carrier absorption is proportional to the square of the wavelength[19], the detector response is superior at longer wavelengths.

Uncooled detection of IR radiation is becoming important owing to the necessity of such detectors in a wide range of applications in the civilian, industrial, medical, astronomical, and military sectors. Cryogenic cooling or high-power-consuming multi-stage thermo-electric coolers (TECs) are not practical for most applications. This research study focused on novel concepts that can be used to develop uncooled IR detectors. Several successful novel concepts (including a pending patent[20]) with experimentally verified results have been already published in peer-reviewed journals[21-28].

In addition to uncooled detector development, this study investigated a GaSb homojunction based terahertz (wavelength range from 30 μm to 1 mm) detector. The use of terahertz radiation as a tool for characterization of materials has been widely demonstrated[29]. Applications can be found in various fields such as medicine[30], industry, security, astronomy, and atmospheric studies. Some examples include cancer/tumor detection, non-destructive testing, toxic chemical detection, and gas sensing[31]. The key advantage of terahertz radiation in these areas is the ability to penetrate and distinguish between different non-metallic materials. This dissertation is arranged into five chapters and, the first chapter gives an introduction by describing the background and purpose of the study as well as the outline of the dissertation.

In the first part of Chapter 2, dye-sensitized near-IR room-temperature photovoltaic photon detectors are presented. Dye molecules bonded to a semiconductor surface can inject carriers to a conduction band by photo-excitation. This process known as dye-sensitization is used for extending the sensitivity of silver halide emulsions[32] and was discovered a long time ago in 1839. More recently (around 1991), dye-sensitization has been adopted to devise solar cells[33]. A near IR sensitive heterojunction *n*-TiO_2/Dye/*p*-CuSCN (where Dye denotes a near-IR absorbing pigment) is developed[21] to examine the possibility of using dye-sensitization for IR detection. Although the responsivity is low and response slow compared to silicon photo detectors, dye-sensitized (DS) detectors would be cost effective, especially for large area devices. They can operate at room temperature and have the advantage of insensitivity to noise induced by band-gap excitations (providing high specific detectivity of $\sim 10^{11}$). Furthermore, the spectral response can be adjusted in the NIR range by choosing the appropriate dye. These results were published in Applied Physics Letters.[21]

The second part of chapter 2 describes *1/f* noise studies on nanostructured semiconductors, DS photon detectors, and solar cells. In general, electronic devices are plagued with *1/f* noise originating from many causes. The most important factors contributing to *1/f* noise in a semiconductor is believed to be generation recombination of carriers and their trapping at defects and impurity sites. Adsorption of moisture and electron acceptor molecules enhances the intensity of *1/f* noise. Amazingly, some molecular species that strongly chelate to the semiconductor surface suppress *1/f* noise owing to passivation of the recombination sites. Thus, in addition to sensitization, the dye adsorbed on the nanocrystallites plays a key role in mitigation of recombinations. For this reason DS heterojunctions could also find application as low noise NIR photon detectors. Experiments conducted with oxide semiconductors (TiO_2, ZnO, SnO_2) indicate that the mode of binding of dyes at specific sites determines the extent to which the recombination and *1/f* noise is suppressed. The transport of electrons in a nanocrystalline matrix is diffusive with a diffusion coefficient D depending on the trapping and detrapping processes. Thus, passivation of trapping sites by the adsorbed dye is expected to increase the response time which can be expressed as $\tau \sim L^2/D$, where L = thickness of the nanocrystalline film. These results were reported in Semiconductor Science and Technology.[28] and Infrared Physics & Technology.[27]

Chapter 3 describes a colloidal quantum dot (CQD) based capacitive-type detector. It is shown that the high electronic polarizability of quantum dots can be utilized to devise photon detectors by embedding quantum dots in a dielectric media to form capacitors. Modulated light generates displacement currents and an expression is obtained for responsivity in terms of the properties of the quantum dot, the dielectric, and the detector geometry. A model detector consisting of PbS quantum dots embedded in paraffin wax is devised to illustrate the principle, giving ~0.6 A/W as an upper limit for the responsivity. As these systems sense only the variations of the light intensity, they could be operated at ambient temperature. This was published in Applied Physics Letters.[25]

Chapter 4 describes spin split-off transition based uncooled IR photon detectors. In the first part of this study, heterojunction interfacial workfunction internal photoemission (HEIWIP) detectors that are not optimized for split-off transitions were used to demonstrate IR response originating from hole transitions between light-/ heavy-hole bands and the split-off (spin-orbit) band. A GaAs/AlGaAs heterojunction with a threshold wavelength of ~20 μm, shows an operating temperature of 130 K for split-off response in the range 1.5 –5 μm with a peak D* of 1.1×10^5 Jones. Analysis suggests that practical devices with optimized parameters are capable of achieving room temperature operation with higher specific detectivity. In the second part of the study a set of three *p*-GaAs/$Al_xGa_{1-x}As$ multiple heterojunction detector structures with free carrier threshold wavelengths of 8, 6 and 4 μm were designed and tested to investigate the best parameters for the spilt-off band detectors. Uncooled and even above room temperature (330 K) satisfactory operation was achieved for the 4 μm free carrier threshold sample with peak responsivity of 0.29 mA/W at 2.5 μm at 300 K. In addition to the expected split-off response, long wavelength response up to 60 μm was observed in this split-off sample. The mechanisms that could generate this type of response are also discussed. These results were reported in two Applied Physics Letters[22,24] and Infrared Physics & Technology[23] article.

In Chapter 5, a GaSb based homojunction interfacial workfunction internal photoemission THz (far-IR, >30μm) detector is presented. Metal-organic vapor phase epitaxy grown *p*-GaSb/GaSb samples show 9.7 A/W peak responsivity and a peak detectivity of 5.7×10^{11} Jones with effective quantum efficiency of 33% at 36 μm and 4.9 K. The detector exhibits a 97 μm (~3 THz) free carrier response threshold wavelength. Results indicate that *p*-GaSb HIWIP detectors are promising candidates as a competitor for current THz detectors. GaSb is not a well developed material system compared to GaAs, Si, or Ge. The GaSb homojunction detector was analyzed as an initial study to test the quality of the material and processing. Even though this detector operated at liquid He temperatures, a design with lower (~15 μm) free carrier threshold should cover 8 to 14 μm atmospheric window region operating at higher temperature (240 K) which can be achieved with TEC. This was reported in Applied Physics Letters.[26]

Appendix A describes the basic detector characterization techniques such as I-V-T, C-V-T, noise, and spectral response measurements. The terms related to the detector characterization (such as specific detectivity, responsivity, noise equivalent power, etc.) and their standard definitions are included in this section.

Several software packages have been developed using Microsoft Visual Basic to automate the detector characterization process and improve quality and efficiency of the measurements. All these programs are incorporated into standard measurement setups in the Optoelectronic Laboratory at GSU. Appendix B describes the main features of these programs and the source code is included at the end of each subsection.

Chapter 2

Dye-sensitized near-infrared room-temperature photovoltaic photon detectors and 1/f noise in semiconductor nanostructures

2.1 DS photon detectors

The development of devices for sensing IR radiation continues to be an important area of investigation because of the application potential in a diverse variety of devices[34-38]. Most widely used IR detectors depend on electron-hole generation in low bandgap semiconductor structures by incident radiation. These detectors exhibit good signal-to-noise performance and very fast response. However, in order to achieve this, the semiconductor components of the detector need cooling. Thermal generation of carriers in a low bandgap semiconductor, impurity and defect mediated recombination or thermally activated intersubband transitions (i.e., in quantum well detectors) are limitations of conventional semiconductor photon detectors. In dye-sensitization (DSN), dye molecules anchored to a semiconductor surface inject carriers into a band leaving dye ions of opposite charge on the semiconductor surface[39]. The transfer of the charge on dye ions to a counter-electrode through a suitable medium (liquid[33] or solid[40,41]) yields a photocurrent having a spectral response (action spectrum) that is commensurate with the optical absorption of the dye. An energetic requirement for DS electron injection is the location of the excited energy level of the dye molecule above the semiconductor conduction band edge. Similarly for hole injection, the ground level of the dye molecule should have a position below the valence band edge. The possibilities of using high bandgap semiconductors insensitive to background thermal noise and the flexibility of spectral response by choice of the dye are advantages of DSN based photon detectors. Furthermore, DSN produces only one type of carrier in the semiconductor volume, therefore the DS photocurrent remains insensitive to recombination at the grain boundaries or impurity sites[39]. These attractive features of DSN have been exploited to construct solar cells using nanocrystalline films of oxide semiconductors[33,39-41]. Dyes absorbing light in the NIR region of the spectrum are commercially available for use as photographic sensitizers or laser dyes. This investigation indicates that some of these dyes sensitize the heterojunctions of the configuration, *n*-type semiconductor/dye/*p*-type semiconductor. A new detector based on nanocrystalline TiO_2 (bandgap = 3.1 eV) film as the dye coated *n*-type substrate and *p*-CuSCN (bandgap 3.6 eV) as the hole collector is fabricated and tested. Results indicate that these systems could be used as wavelength tailorable, room-temperature, low cost, IR detectors.

2.1.1 DS IR detector fabrication

Nanocrystalline films of *n*-TiO_2 were coated on fluorine doped conducting tin oxide (CTO) glass plates (1×2 cm^2, sheet resistance = 15 ohm/square) by the method reported earlier[42]. Titanium isopropoxide (5 ml) mixed with propan-1-ol (20 ml) + glacial acetic acid (5.5 ml) hydrolysed by gradual addition of water (5 ml) is grounded with Degussa P25 TiO_2 powder (median particle size ~50 nm). The resulting paste was painted over cleaned CTO glass plates (NaOH + propan-2-ol, followed by water) heated to 150ºC, sintered in air at 450ºC for 10 min.

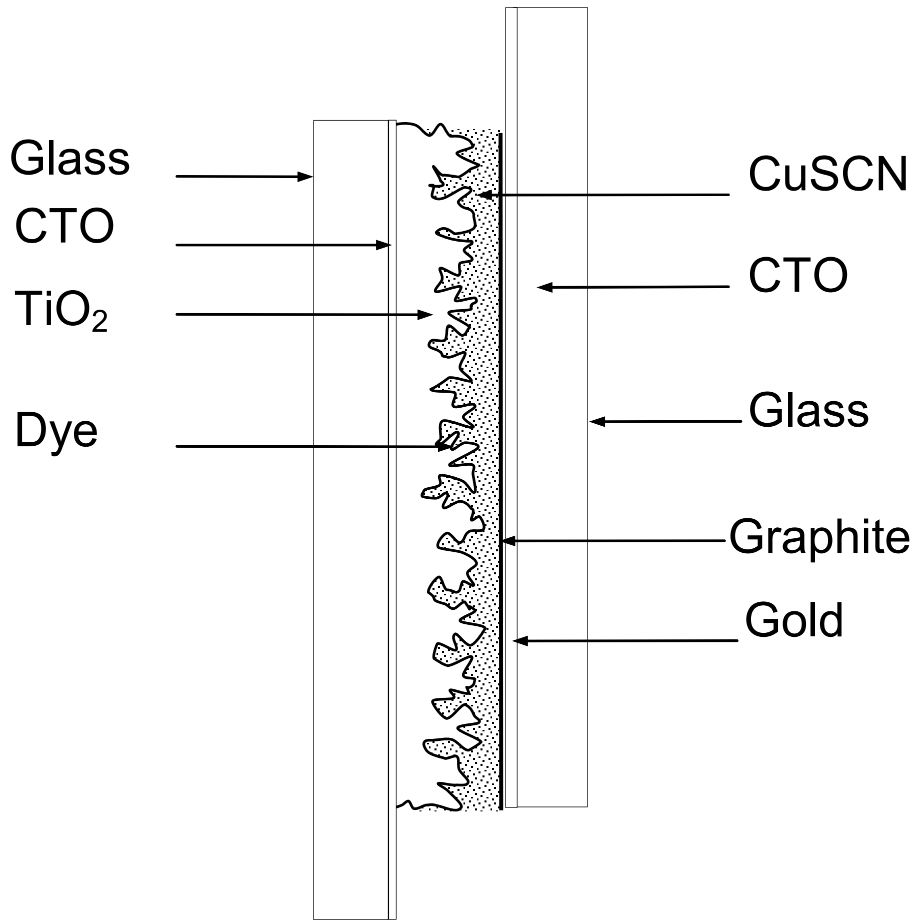


Figure 2.1 Cross-section of the DS NIR detector. Here CTO denotes the conducting tin oxide layer that is transparent to VIS and NIR. The schematic exaggerates the TiO_2 surface in order to clearly show the roughness of the layer.

After removing the loose crust of TiO_2 on the surface, the process was repeated until a film of ~10 μm thickness is formed. Dyes used in this investigation were anionic dyes IR 783, and IR 820 and cationic dyes IR 792, and IR 1040 purchased from Aldrich (numbers indicate the peak absorption wavelength of dye solution in nm). The anionic dyes readily anchor to the TiO_2 surface and were coated by immersing the TiO_2 film in a solution of the dye in 90% ethanol. The cationic dyes not directly anchoring to the TiO_2 surface were deposited by pre-adsorption of another molecular species, usually an anionic compound[43]. Thereafter, the film was rinsed with ethanol and immersed in a solution of the cationic dye. The anionic compounds used for this purpose were bromopyrogallol red (BPR) mercurochrome (MC) and IR820.

The heterojunction *n*-TiO_2/Dye/*p*-CuSCN was fabricated by deposition of *p*-CuSCN on dyed surface from a solution of CuSCN in propyl sulfide[42]. The outer CuSCN surface was coated with graphite and a gold plated CTO glass plate pressed onto the graphite surface served as the back contact of the detector. A schematic diagram indicating the construction of the device is shown in Figure 2.1. Photocurrent spectral responses of the cells were recorded using a monochromatic-light, chopper lock-in amplifier system. Light intensities were measured with a calibrated silicon diode. (See Appendix B.2 for experimental setup and details about the in-house data collection software program.)

2.1.2 Experimental results and discussion

Nanocrystalline TiO_2 films prepared by the method described above have effective surface areas 200-300 times the geometrical area of the film[40]. Although the dye covers the TiO_2 surface at the monolayer level, the large roughness factor of the film increases the light absorption cross-section.

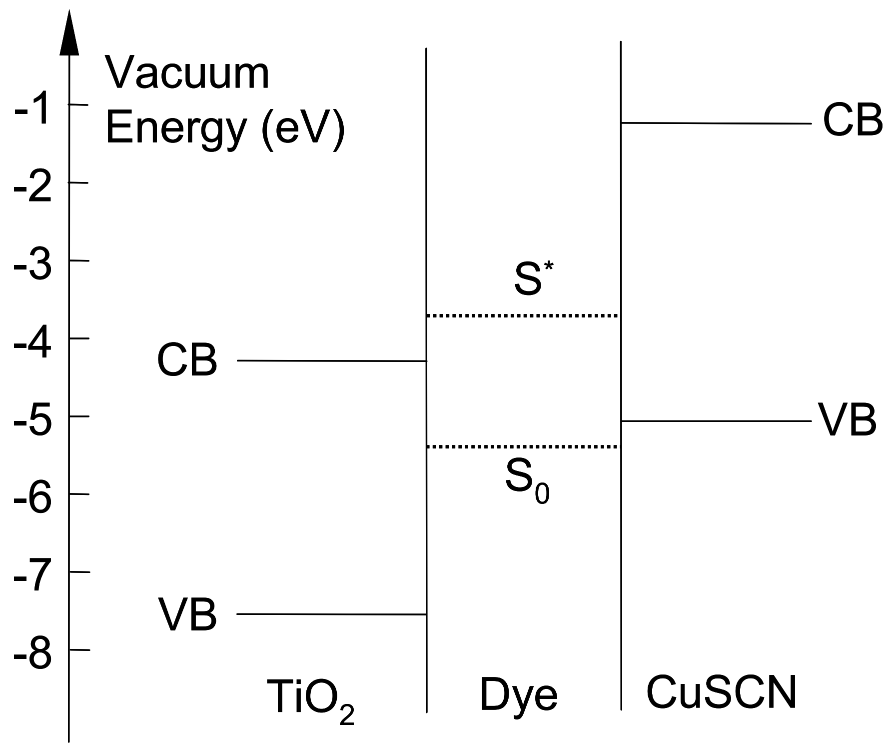


Figure 2.2 An energy level diagram showing the band edge positions of *n*-TiO_2, *p*-CuSCN and ground (S_o – positioned arbitrarily) and excited (S*) states of the dye.

A schematic energy level diagram showing the band edge positions of TiO_2, CuSCN and ground (S_0) and excited (S^*) levels of the dye is presented in Figure 2.2. The photovoltaic effect of the heterojunction originates via the following charge transfer scheme. Excited dye molecules (***Dye****) inject electrons and holes to the conduction and valence bands (CB and VB) of *n*-TiO_2 and *p*-CuSCN respectively, i.e.,

$$\begin{aligned} &h\nu + Dye \rightarrow Dye^* \\ &Dye^* \rightarrow Dye + e^-(\text{CB-TiO}_2) + h^+(\text{VB-CuSCN}) \end{aligned} \tag{2.1}$$

The relative positioning of the ground (S_o) and excited (S^*) levels of the dye and band edges of the two semiconductors (as shown in Figure 2.2) energetically permit the above charge transfers. The positions of the band edges of TiO_2 and CuSCN are known[44] and agree with the relative band offsets as shown in Figure 2.2. However, the positions of S_o and S^* on the absolute scale were not measured, which normally requires a cyclic voltammetric determination of the position of S_o with respect to a standard redox couple[45]. The observation of sensitized photocurrents with IR 783, IR 820, IR 792 and IR 1040 shows that these dyes have ground and excited level positions satisfying the above conditions. The spectral responsivities of the detectors sensitized with IR dyes are shown in Figure 2.3. The red-shifts of the peak positions in action spectra are caused by bonding of dyes to the semiconductor surface. The responsivities at the peak absorption wavelengths summarized in Table 2.1, are nearly two orders of magnitude smaller than that of silicon detectors[46]. The peak specific detectivity D^* of the system *n*-TiO_2/MC-IR792/*p*-CuSCN was found to be 9.5×10^{10} cm $Hz^{-1/2}$ W^{-1} at 812 nm. (noise characteristics measured using the low noise SR560 preamplifier and the SR785 fast Fourier transform Dynamic Signal Analyzer; see Appendix B.3.1 and A.4.1 for experimental setup and calculation method)

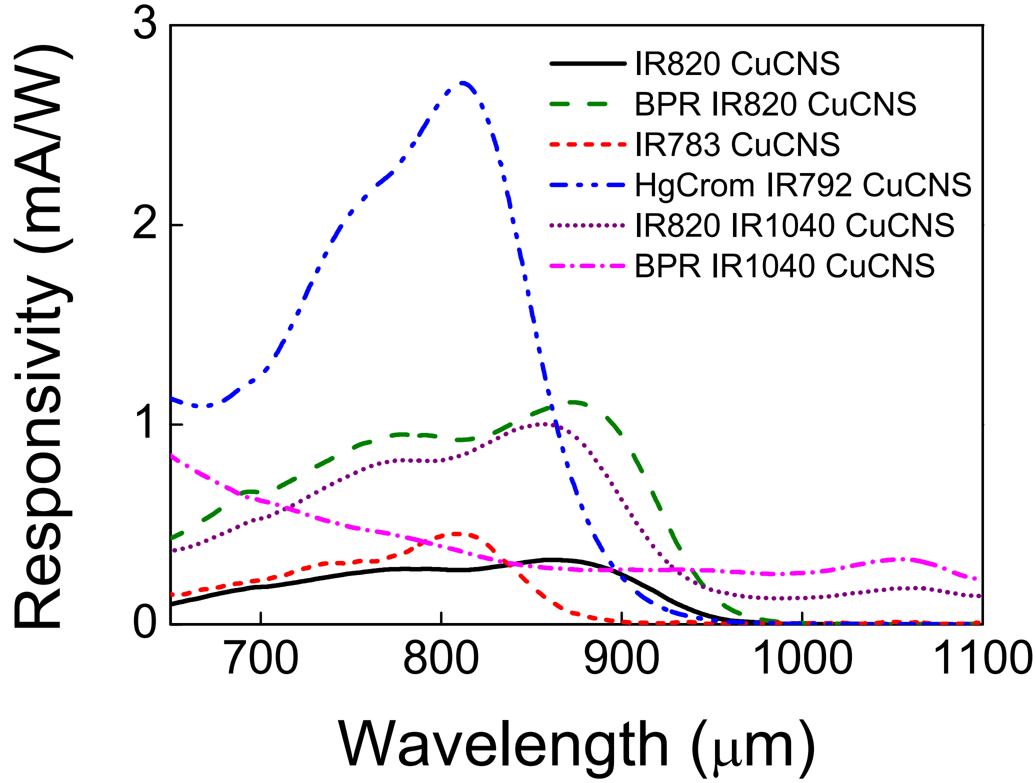


Figure 2.3 Spectral responsivity of the heterojunctions (a) *n*-TiO_2/IR820/*p*-CuSCN (b) *n*-TiO_2/BPR-IR820/*p*-CuSCN (c) *n*-TiO_2/IR783/*p*-CuSCN (d) *n*-TiO_2/MC-IR792/*p*-CuCSCN (e) *n*-TiO_2/IR820-IR1040/*p*-CuSCN (f) *n*-TiO_2/BPR-IR1040/*p*-CuSCN

Poor photon-to-photocurrent conversion efficiency could be a result of low probability of injection of carriers by the excited dye molecule. Photophysical properties of the dye, the relative energetic positions of ground and excited states with respect to the band edges, density of states in the bands (i.e., electrons in the CB of TiO_2 and holes in the VB of CuSCN) and mode of anchoring of the dye molecule to the TiO_2 surface determine the injection rate. When photocurrent transients of the detector sensitized with IR 792 were examined, the rise and fall time constants turned out to be of the order 5 and 3 ms, respectively. Clearly, the response is slower than that of conventional semiconductor detectors. This behavior of the device is a consequence of the slow diffusive transport of electrons in a film consisting of TiO_2 nanocrystallites interconnected to each other by sintering. The response time τ of the detector can be expressed in the form,

$$\tau \approx \frac{L^2}{D} \tag{2.2}$$

where L = film thickness, D = diffusion coefficient (which depends on film morphology and intrinsic material properties of the two semiconductors). Thus, the response time happens to be highly sensitive to the film thickness, i.e., a reduction in film thickness leads to faster response. However, the responsivity of the detector determined by the light absorption cross-section of the dye coated nanocrystalline film varies linearly with the film thickness. Optimization would be possible to meet the requirements of specific practical applications.

2.1.3 Conclusion and future possibilities

This investigation demonstrates that dye–sensitization of high bandgap semiconductors can be utilized to devise NIR detectors. They have the advantage of not being sensitive to thermal noise and radiation that initiate bandgap excitations.

Table 2.1 Responsivities (R) of the detectors at peak absorption wavelengths (λ_{max}) of different sensitizers.

Dye	**λ_{max} (nm)**	**R (mA/W)**
IR820	866	0.3
BPR-IR820	876	1.1
IR783	808	0.4
MC-IR792	812	2.7
IR820-IR1040	858	1.0
BPR-IR1040	1056	0.3

As expected, the specific detectivities of DS detectors are found to be quite high despite low responsivity. The specific detectivity D^* of the system n-TiO_2/MC-IR792/p-CuSCN was found to be 9.5×10^{10} cm $Hz^{-1/2}$ W^{-1} at 812 nm. As shown in Figure 2.4, the variation of the dark current noise spectral density $S(f)$ with frequency at room-temperature (292 K) is low and largely frequency independent beyond a frequency of ~1 kHz. High bandgap semiconductors remain less susceptible to stochastic intermittency of thermal excitations that contribute to $1/f$ noise[47,48]. The peak spectral response of these devices can be readily adjusted by choice of the dye and tuning by structural modifications to the dye molecule. DS NIR detectors would be very cost effective devices because they are based on polycrystalline semiconductor materials and the processing does not involve vacuum technology. Instead of CTO glass, TiO_2 film could be deposited on a flexible plastic substrate[49]. This will allow detector to be fabricated even on a curved-surface. DS NIR detectors can be extended to panchromatic detection utilizing dye-multilayer structure[44] or pair of cationic and anionic dyes[50,51] since this idea is already demonstrated on DS solar cells. However, in the present form the responsivities are of the order 10^{-3} A/W (Table 2.1), i.e., two orders of magnitude smaller than familiar semiconductor detectors. Extensive effort is underway for improvement of the efficiencies of DS solar cells. The outcome of these investigations could pave the way for the development of highly sensitive DS IR photon detectors. Combination of IR dyes with visible sensitizers may also enhance the efficiencies of DS solar cells. In fact, the heterojunctions that incorporate BPR or MC in addition to the IR dye are also sensitive to the visible spectrum.

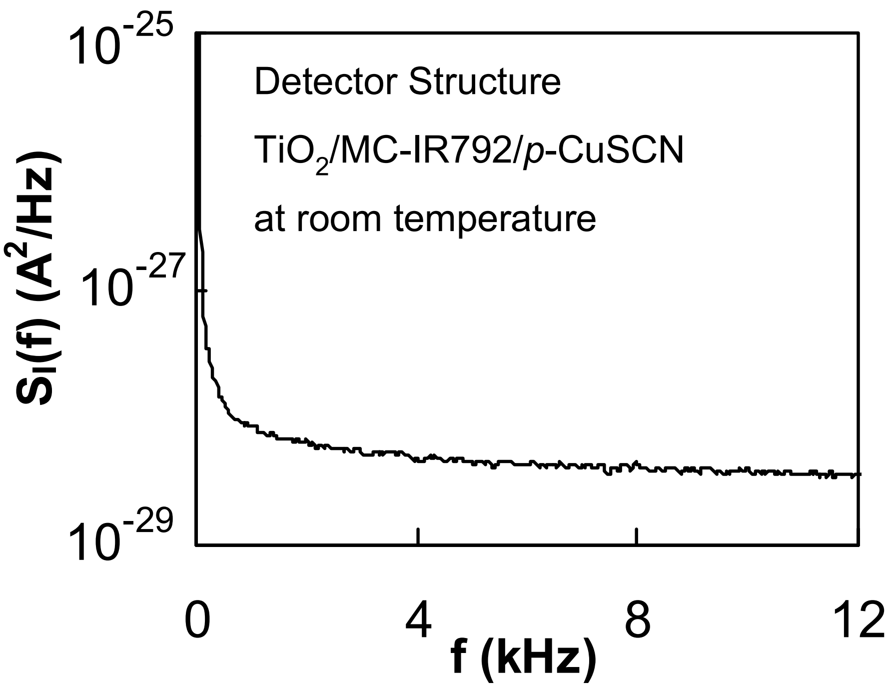


Figure 2.4 Power spectral density of the dark current noise of the hetrojunction *n*-TiO_2/MC-IR792/*p*-CuSCN.

2.2 1/f noise in DS photon detectors and photovoltaic cells

2.2.1 Introduction

The lowest possible level of noise provides the highest detectivity of the photon detectors. Therefore, it is very important to understand the cause of noise and techniques of its suppression. Dye-sensitized near infrared photon detectors (DSNIRPDs) described in Chapter 2 and DS solar cells (DSSCs) based on nanocrystalline high bandgap oxide semiconductors receive much attention as cost effective alternatives to the conventional photon detectors and solar cells made from low bandgap semiconductors. The functioning of these devices depends on DSN, i.e., a process by which a photo-excited dye molecule anchored to a semiconductor surface injects an electron to the conduction band forming a dye cation[33,52]. The positive charge on the dye cation can be transferred to a suitable hole collecting medium in contact with the dyed nanocrystalline surface to form heterojunction of the configuration *n*/Dye/*p* , where *n* denote the nanocrystalline *n*-type semiconductor, and *p* denote hole collecting medium. In one version of DSSCs, the hole-collecting medium is a redox electrolyte[33,52] (also known as Gratzel type solar cells (GDSSCs)) which efficiently collects the positive charge on the dye cations formed during electron injection, whereas in the other version *p* is a hole collecting solid material[40,53] (i.e., a *p*-type semiconductor of appropriate band positions). The electron injected into the *n*-type material and the dye cation on its surface undergoes recombination. However, the rate of recombination happens to be several orders of magnitude slower than the rate of injection and this is one of the reasons why the DSNIRPDs and DSSCs function, converting light to electricity, at reasonably high quantum efficiencies[52]. The dye coverage on the semiconductor surface needs to be maintained at the monolayer level to avoid insulation by thick dye layers and to prevent rapid quenching of the excited molecules by mutual interaction. Although the dye coverage is at the monolayer level, the large roughness factor (~

1000) of the oxide semiconductor greatly increases the light absorption cross-section. As DSN involves carrier transfer to one band, bulk recombinations are absent, but the large effective area of the interface makes DSNIRPDs and DSSC susceptible to surface recombination. However, DSSCs deliver quantum efficiencies exceeding 85% and energy conversion efficiencies around 10% are readily achievable[33,52], indicating that recombination does not occur at the expected rate. Trapping and detrapping of carriers at the surface of semiconductors generate *1/f* noise[54,55]. The interactions of carriers with both bulk and surface trap sites contribute to *1/f* noise. In nanocrystalline semiconductors where the surface to bulk ratio is exceptionally large, the trapping-detrapping of carriers at surface states makes the most significant contribution to *1/f* noise. A nanocrystalline surface is heavily populated with defects and adsorbed species that act as trapping and recombination sites. This chapter describes the observations of 1/*f* noise in bare and dye coated nanocrystalline films of TiO_2. It is found that in bare nanocrystalline TiO_2 films, adsorbed molecular species that produce electron acceptor surface states (e.g., water vapor, iodine) induce 1/*f* noise. However, in dye coated TiO_2 films, the passivation of active sites on the TiO_2 surface by the dye suppresses the generation of 1/*f* noise.

2.2.2 Frequency dependent noise

The electric current through a conductor at a constant bias voltage and temperature undergoes fluctuations around a mean.[54-58] In addition to the frequency *(f)* independent thermal and shot noises, conducting materials and electrical devices also display a noise whose spectral power density exhibits a *1/f* dependence[54-58] (more generally a $1/f^{\delta}$ variation where δ is a positive index, generally close to unity). Under thermostatic conditions the current passing through a nanocrystalline semiconductor films exhibits *1/f* noise according to Hooge's formula[59]

$$S(f) = \frac{AI_0^2}{f^{\delta}} \tag{2.3}$$

where I_o= mean value of the fluctuating current $I(t) = I_o + X(t)$, with $<X(t)>=0$ and

$$S(f) = lim_{T\to\infty}\left(\frac{1}{2T}\left|\int_{-T}^{T} X(t)e^{-2\pi ift}dt\right|^2\right). \tag{2.4}$$

The constant A in (1) which measures the level of noise, generally takes the form[59] $A = \Upsilon/N$, where N = the total number of free charge carriers in the sample and the parameter Υ is referred to as the Hooge's constant. Interaction of carriers with defects, surface states and associated events such as recombination and trapping-detrapping are believed to be the major causes of *1/f* noise in semiconductors.[54-60] Consequently, *1/f* noise measurements receive considerable attention as a tool for characterization and understanding the nature of carrier relaxation processes in semiconductors.[55,56] In electronic measuring devices, the lowest possible level of *1/f* noise is desired to improve the detectivity. Also, a high *1/f* noise level usually indicates an underlying dissipative process[61]. Both bulk[54,55,60] and surface[54,55,62] interactions of carriers contribute to *1/f* noise in semiconductors. Because of the large surface to volume ratio, surface effects are expected to be a major cause of *1/f* noise in nanocrystalline semiconductors. Furthermore, the slow charge transport in nanostructured materials has been attributed to trapping and detrapping of electrons in intraband-gap surface states.[63-65]

2.2.3 Sample preparation and experimental setup for noise measurements

Nanocrystalline films used for measuring the noise in the electrical current were prepared as described below. A scribe (~ 8 μm thick) is drawn on the surface of a FTO glass plate (1.5 × 2 cm^2,

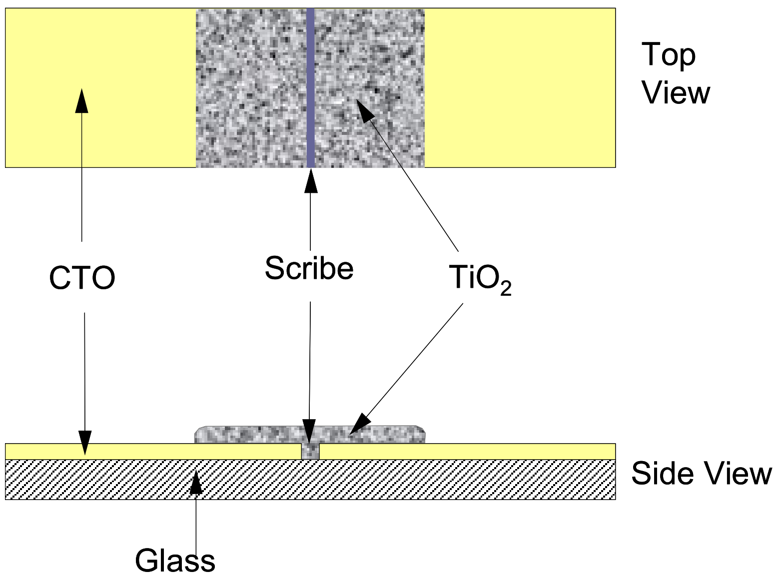


Figure 2.5 A schematic diagram illustrating sample geometry used for the noise measurement. The electrical connectivity of a conducting tin oxide (CTO) glass sheet (1.5 × 2 cm2, sheet resistance 12 ohm/square) was broken by scribing a line (~8 µm thick) with a diamond pointer. A layer of colloidal TiO2 (thickness ~12 µm) was screen printed above the scribe and sintered at 430 ºC for 30 minutes.

sheet resistance 12 ohm cm^{-2}) with a fine diamond point to break the electrical connectivity. A layer of colloidal TiO_2 (thickness ~ 10 μm) was screen printed above the scribe and the plate is sintered at 430° C for 30 minutes to form the sample geometry[66] represented in Figure 2.5. In some experiments, films were coated with BPR or cis-dithiocyanatobis (4-4'-dicarboxylic acid-2, 2'Ru[II]) (N3) by soaking them in alcohol solutions of these dyes (~2.5 × 10^{-3} M). The experimental setup used for the noise measurements is shown in Figure 2.6. The sample placed in a chamber with facilities for evacuation and heating was connected in series with a resistor R = 56 kΩ and an 18 V battery. A low noise preamplifier amplified the voltage fluctuation across the sample and a power spectrum analyzer (Stanford Research SR785 Fourier Transform Dynamic Signal Analyzer) computed *S(f)* and plotted its variation with *f*.

2.2.4 Results and discussion

When the chamber is filled with air at atmospheric pressure and relative humidity (RH) 27% (23 °C), the noise power spectrum (i.e., plot of *S(f)* vs. *f*) clearly displayed frequency dependent noise (FDN). On evacuation of the chamber, FDN persisted but at lower intensity. However, if the film is heated to ~90 °C during evacuation, FDN disappeared completely even after cooling the sample to room temperature under vacuum (10^{-6} Torr) and did not reappear when the chamber was filled with dry N_2 or O_2 (Figure 2.7 curve (a)). Generation of FDN resumed when water vapor was introduced into the chamber. The curve (b) of the Figure 2.7 shows the noise spectrum of the sample in a nitrogen atmosphere at RH ~70%. The above observations show that the factor which contributed to FDN when the sample was in air was also moisture. When the experiment was repeated replacing water with acetonitrile, FDN was not detected. FDN was once again observed with introduction of iodine into the chamber filled with N_2 (Figure 2.7, curve (c)).

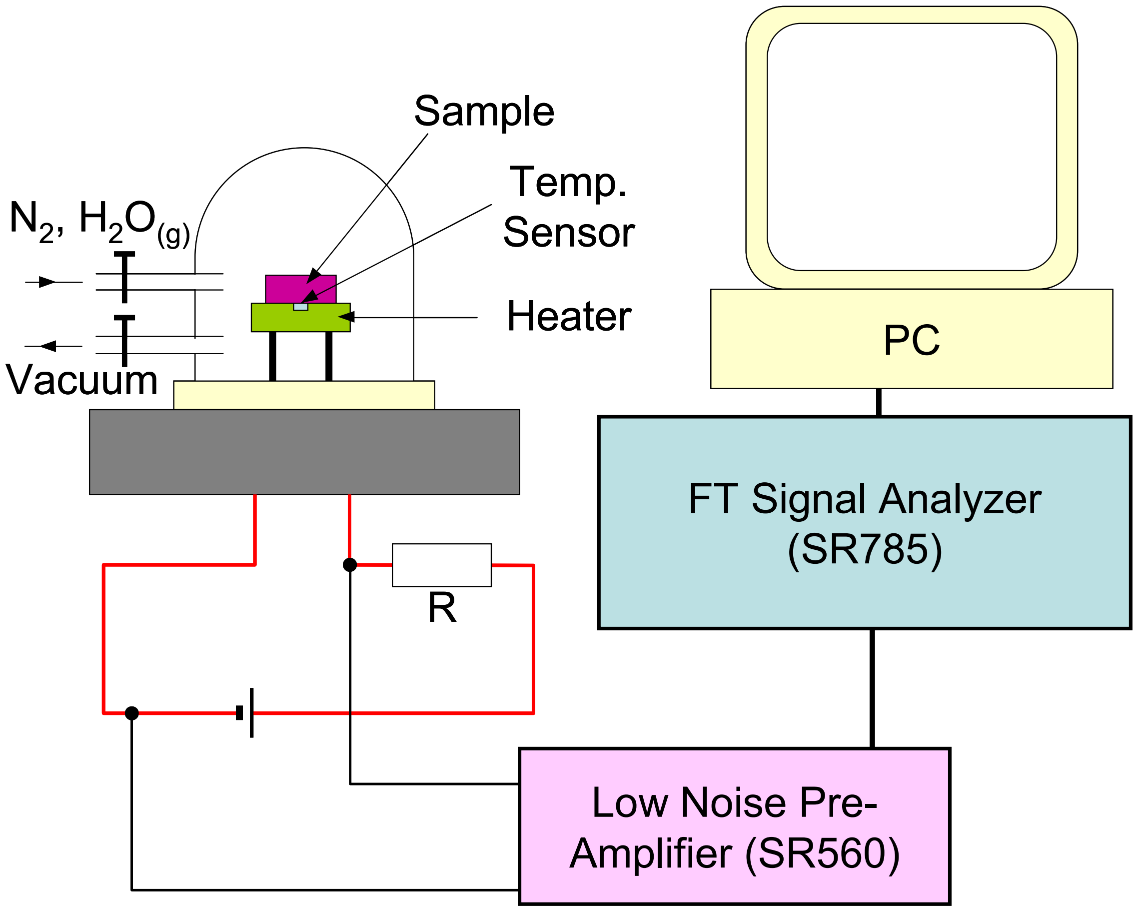


Figure 2.6 The experimental set up used for noise measurements. The sample is mounted in an enclosure provided with facilities for evacuation, introduction of different gasses/vapors, and heating, and the sample is connected in series with a 56 kΩ resistor (R) and 18 V battery pack. A low noise preamplifier amplified the voltage fluctuations across the sample, and a power spectrum analyzer was used to obtain the spectral power density S(f).

It was interesting to note the difference in behavior of bare and dye coated TiO_2 films. Here, iodine was ineffective in creating FDN and the noise power density (f independent) remained unaltered in the presence of iodine vapor in the chamber (Figure 2.7 curves (d), (e), (f), (g)). Again the influence of moisture in generating FDN is less in dye coated TiO_2 films as measured noise levels here are smaller compared to that of the bare TiO_2 films. Table 2.2 gives the values of the parameters A, δ and Υ when the noise spectra of different systems are fit with the formula (2.3). The magnitude of the parameter A for a sample of given geometry (i.e., film thickness, length and breadth of the scribe) measures the level of $1/f$ noise. In the absence of FDN, the background noise (Figure 2.7 curves (a), (d) and (e)) originates from thermal and environmental influences and does not follow formula (2.3). Nevertheless, for comparison Table 2.2 gives the values of A and Υ for the frequency independent situations as well, setting $\delta = 0$ in formula (2.3). Hooge's constant Υ is estimated using the literature value[67] for carrier density of TiO_2 ($\sim 1\times10^{17}$ cm^{-3}) and volume of TiO_2 in the scribe = 7.2×10^{-3} cm^3 (estimated from the knowledge of the volume of TiO_2 in a large size film). The above experiments indicate that the electron acceptor states produced by adsorbed species generate *1/f* noise in nanocrystalline TiO_2 films. Water is known to dissociatively adsorb on the TiO_2 surface forming H and OH fragments[68]. The acceptor OH radicals act as electron trapping sites. Similarly, iodine adsorbed on the TiO_2 surface could also trap electrons. Trapping and detrapping of electrons at the surface states is known to be a cause of $1/f$ noise in semiconductor materials.[54] This process becomes exceedingly important for nanostructured films because of their large surface to volume ratio. In a dyed TiO_2 film, water and iodine absorption sites get passivated by surface chelation of the dye molecules and therefore $1/f$ noise is suppressed.

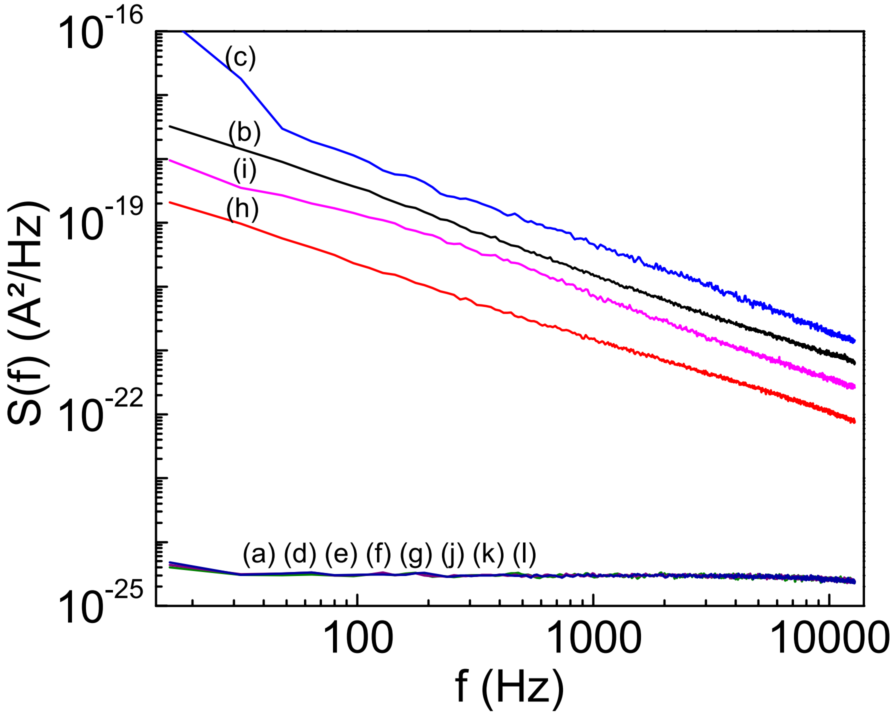


Figure 2.7 Noise spectra at 23 °C of, (a) bare TiO2 film in N_2. (b) bare TiO_2 film in N_2 at RH ~70 %. (c) bare TiO_2 in a N_2 saturated with I_2 vapor. (d) TiO_2/BPR in N_2. (e) TiO_2/N3 in N_2. (f) TiO_2/BPR in N_2 saturated with I_2. (g) TiO_2/N3 in N_2 saturated with I_2. (h) TiO_2/BPR in N_2 at RH ~70 %. (i) TiO_2/N3 in N_2 at RH ~70%. (j) TiO_2 (vacuum) (k) TiO_2/BPR (vacuum) (l) TiO_2/N3 (vacuum).

Table 2.2 The values of parameters A, δ and γ for different systems obtained by fitting noise data with formula (2.3), biasing voltage = 18 V, and $I_o = 3.2\times10^{-4}$ A.

sample	***δ***	***A***	**ϒ**
TiO_2 (vacuum)	0	4.4×10^{-18}	3.2×10^{-7}
TiO_2 (N_2)	0	4.4×10^{-18}	3.2×10^{-7}
TiO_2/BPR (vacuum)	0	4.4×10^{-18}	3.2×10^{-7}
TiO_2/N3 (vacuum)	0	4.4×10^{-18}	3.2×10^{-7}
TiO_2/BPR (N_2)	0	4.4×10^{-18}	3.2×10^{-7}
TiO_2/N3 (N_2)	0	4.4×10^{-18}	3.2×10^{-7}
TiO_2 (N_2 ,RH = 70%)	1.25	8.8×10^{-10}	6.3×10^{1}
TiO_2/BPR (N_2, RH= 70%)	1.15	4.4×10^{-11}	3.2×10^{0}
TiO_2/N3 (N_2, RH = 70%)	1.30	5.7×10^{-10}	4.1×10^{1}
TiO_2 (N_2, saturated I_2 vapor)	1.37	5.8×10^{-9}	4.2×10^{2}
TiO_2/BPR (N_2, saturated I_2 vapor)	0	4.3×10^{-18}	3.1×10^{-7}
TiO_2/N3 (N_2, saturated I_2 vapor)	0	4.4×10^{-18}	3.2×10^{-7}

When the RH in the chamber is high (≥40%), dye bonded to TiO_2 surface seems to desorb partly, allowing water adsorption. This explains the observation that in an environment of high H_2O vapor pressure, dyed TiO_2 films also exhibit $1/f$ noise.

2.2.5 Conclusion

The observations presented above have profound implications for the performance of GDSSCs[33] and DSNIRPDs[21]. In fact they explain why the GDSSCs and DSNIRPDs work so well. Although the substrate through which the electrons move is highly populated with defects, the adsorbed dye passivates these surface recombination centers. As the trapping-detrapping process at surface states is relieved by dye chelation, the dye absorption also facilitates transport of electrons across the nanocrystalline matrix. In fact, it has been noted that the diffusion coefficient of electrons in dyed nanocrystalline films is higher than that of bare films.[69] If there are voids in the dye monolayer on TiO_2 surface, interaction of iodine at the points of exposure of TiO_2 would be detrimental to the functioning of the GDSSCs. The trapping of electrons by iodine adsorbed at the voids in the dye layer will cause recombination. Partial desorption of the dye, followed by iodine adsorption seems to be a major cause of recombination loss in the GDSSCs. It is interesting to note that successful attempts have been made to synthesize dyes that resist desorption to the electrolyte.[70] Our experiments also indicate the importance of hydrophobic[52] and water free electrolytes. Noise measurement also offers a way of characterizing dye adsorbed nanostructured semiconductor films. The dyes used in this study, N3 and BPR, strongly complex onto the TiO_2 surface via carboxylate and hydroxyl ligands respectively, eliminating the surface hydroxyl groups.[71] The $1/f$ noise studies would be of value in understanding other modes of interactions of molecules with nanocrystalline semiconductor materials.

Chapter 3

Displacement currents in colloidal quantum dots embedded dielectric media: A method for room temperature photon detection

3.1 Introduction

The unique properties of low dimensional semiconductors offer opportunities for application in almost all areas of electronics[72-76]. Many concepts have been extensively studied identifying potential applications. Quantum dots (QDs) are used in photon detection[72-74,77], especially the near infrared (NIR) and infrared (IR) regions of the spectrum[78-81]. Utilizing size quantized band–gap modulation, QDs of low effective carrier mass semiconductors can be sensitized to the electromagnetic spectrum from ultraviolet (UV) to far IR. Photovoltaic and photoconductive detectors have been made from QDs blended in conducting polymers[82-84]. In photovoltaic detectors, interpenetrating networks of polymer and QDs communicate with two electrodes. The excitons generated by the incident photons decompose at the interface into electron-hole pairs which separate into the two regions generating a photocurrent and a photovoltage. Photoconductive type operates by derivation of a current by an external voltage via movement of carriers across the polymer medium. Properties of individual QDs are greatly obscured by clustering and aggregation, and also obtaining electronic contacts to QDs would not be an easy task. The photoconductive detectors, where the QDs are homogeneously impregnated into a solid substrate avoid the above problem. This study shows that by embedding QDs in a film of high dielectric material to form a capacitor, the displacement current generated by modulated light can be used as a signal to detect photons. As in pyroelectric detectors[85], this technique has the advantage that only the intensity modulated light generates signals enabling room temperature operation for sensing IR radiation. Again as in photoconductive QD-polymer detectors, their fabrication does not require electrical connections to the QDs. A capacitor with PbS colloidal quantum dots (CQDs, Colloidal Quantum Dots are synthesized from precursor compounds dissolved in solutions) embedded in paraffin wax (Figure 3.1 a) was used to illustrate the principle.

3.2 PbS colloidal quantum dot (CQD) preparation and detector fabrication

PbS CQDs embedded paraffin wax films were prepared by the following method. Water insoluble lead oleate was synthesized by mixing equal volumes of lead acetate (0.1 M) and sodium oleate (0.2 M) solutions. The white precipitate of lead oleate was separated, washed with water and dried in a vacuum. A weighed amount of lead oleate was dissolved in molten purified paraffin wax (melting point ~ 64º C, dielectric constant = 2.4) and a thin layer of wax was spread on the surface of a conducting glass plate (1.5×1 cm^2). After solidification of wax, the plate was inserted into a N_2 atmosphere containing H_2S (~ 20% by volume) and propan-2-ol vapor and left there for 2 hrs. H_2S diffuses into the wax impregnating it with oleic acid capped PbS CQDs. Presence of the vapor of a slightly polar liquid (propan -2-ol) facilitates this diffusion controlled reaction. The absorption spectrum of the film is shown in Figure 3.1 (b) and the spectrum of the film material dissolved in hexane was also recorded. These spectra suggest a polydispersion of PbS CQDs in wax with a median diameter of ~ 8 nm. To form the capacitor, the plate is warmed to melt the wax and another conducting glass plate posed above to cover an area of 1 cm^2 and fill the capillary space with molten wax. When the wax solidifies, the plates hold together and leads are connected to the two protruding ends of the conducting glass plates (see Figure 3.1 (a)). The measured capacitance (200 pF) of the system was nearly of the same order as that of a capacitor of same dimensions (area = 1 cm^2, thickness = 10 μm) with a film of pure wax and the resistance exceeded 1 GΩ.

3.3 Detection mechanism, results, and discussion

Calculations indicate that QDs have several orders of magnitude larger polarizabilities than that of atoms and molecules[86,87]. This has been confirmed by quantum confined Stark effect[88,89] and measurement of the electronic polarizability of optically generated excitons in quantum dots[90]. Using effective medium theory[90,91], the electric susceptibility χ of the QD- dielectric composite can be expressed as,

$$\chi = \varepsilon_0 n \kappa \alpha \tag{3.1}$$

Where

$$\kappa = \frac{9\varepsilon^2}{(\varepsilon_{QD} + 2\varepsilon)^2} \tag{3.2}$$

and where n = number density of excitons per QD, α = exciton polarizability, ε = dielectric constant of the composite and ε_{QD} = dielectric constant of the QD material. Hence, if a capacitor of thickness s consisting of N QDs of given size per unit volume is placed in a constant electric field E, the displacement current density $dD/dt = J(t)$ originating from time variation of n can be written as,

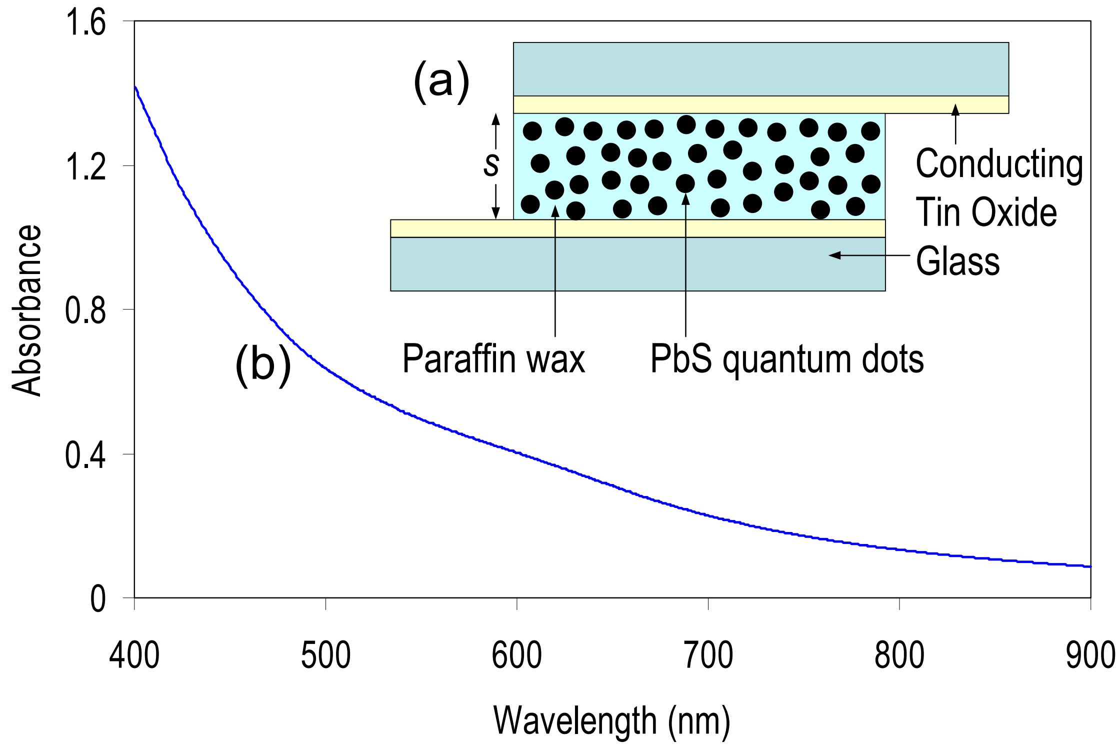


Figure 3.1 (a) Absorption spectrum of the suspension of PbS colloidal quantum dots (CQD). (b) A cross section of the CQD embedded detector experimentally tested. The thickness of the CQD embedded paraffin wax layer is s ~ 10 μm.

$$J(t) = \frac{dD}{dt} = sN\varepsilon_0\kappa\alpha E\frac{dn}{dt}\,. \tag{3.3}$$

For simplicity, the displacement current response of the system was analyzed by assuming that the photon flux incident on the capacitor modulates as, $I = I_0\ (1 + sin\omega t)$. Thus the rate of exciton generation is given by,

$$\frac{dn}{dt} = \phi I_0(1 + sin(\omega t)) - kn \tag{3.4}$$

where ϕ is the quantum efficiency of exciton formation and k is the exciton recombination rate constant. Solving Eq.(3.4) for n and combining with Eq.(3.3), the displacement current density can be expressed in the form,

$$J(t) = \frac{\phi NEI_0\omega\kappa\alpha\varepsilon_0}{\sqrt{\omega^2 + k^2}}\left(sin(\omega t + \delta) + \frac{ke^{-kt}}{\sqrt{\omega^2 + k^2}}\right) \tag{3.5}$$

where tan $\delta = k/\omega$ and the initial condition is $J(0) = 0,\ t = 0.$ When the transient in Eq. (3.5) decays, the detector output current density simplifies to

$$J(t) = \frac{\phi NEI_0\kappa\alpha\varepsilon_0\omega sin(\omega t + \delta)}{\sqrt{\omega^2 + k^2}}\,. \tag{3.6}$$

Setting $E = V/s$ (V = applied voltage bias), the current (A/W) and voltage (V/W) responsivities of the detector can be obtained as follows (R_L, R_d as indicated in Figure 3.2).

$$R = \frac{\sqrt{2}\phi(\nu)\omega VN\kappa\alpha\varepsilon_0}{h\nu s\sqrt{\omega^2 + k^2}} \quad R_V = \frac{\sqrt{2}\phi(\nu)\omega VN\kappa\alpha\varepsilon_0}{h\nu s\sqrt{\omega^2 + k^2}} \times \frac{R_L \times R_d}{R_L + R_d}\,. \tag{3.7}$$

The circuit used for the measurement of the photoresponse, including the equivalent circuit of the detector is shown in Figure 3.2. The effect of the parasitic capacitance in the circuit is not very strong at low chopping frequencies (i.e., $\omega < [R_dC_d]^{1/2}$). The plot of responsivity vs. wavelength at different bias voltages (8.7 V to 40 V) and a fixed chopping frequency of 57 Hz is shown in Figure 3.3. Despite the simplicity of the system, the response extends from 400 - 1100 nm.

The formula (3.7) enables calculation of the intrinsic responsivity of the detector in terms of the properties of the QDs, the embedding dielectric medium, the thickness of the film and the bias voltage. Responsivity depends on several factors, i.e., size and the number density of QDs, their polarizability, the thickness of the capacitor and the applied bias. The response increases linearly with the applied bias according to Eq. (3.6) up to ~ 20 V and deviation from linearity is seen thereafter. A higher bias will also increase the noise and at biasing values exceeding the field ionization threshold, the motion of dissociated electrons and holes will also contribute to the displacement current, causing a deviation from the linear variation. The Eq. (3.7) involves the recombination rate constant k which is sensitive to the structure of the dot (i.e., capping, size and shape) and its environment. Under optimum conditions (i.e., absence of trapping of the carriers) the exciton recombination is slow (~ ms range) and the condition $\omega >> k$ should be satisfied. The constant k can be readily estimated from photocurrent transients (when radiation of constant intensity is interrupted, the transient signal takes the form $sN\kappa\alpha E\phi I_o e^{-ikt}$) and the value obtained is of the same order of magnitude (i.e., ms). For optimization of the responsivity, the quantum dots need to be densely packed keeping the film thickness s comparable to the mean free path of photons. However, in general conductor-insulator composites exhibit percolation thresholds[92] when the packing fraction of the conducting material exceeds a critical value $N=N_C$. In the present system such a threshold would depend on the size of the quantum dots. However, determination of this threshold for monodispersions of PbS CQDs in paraffin was not succeeded.

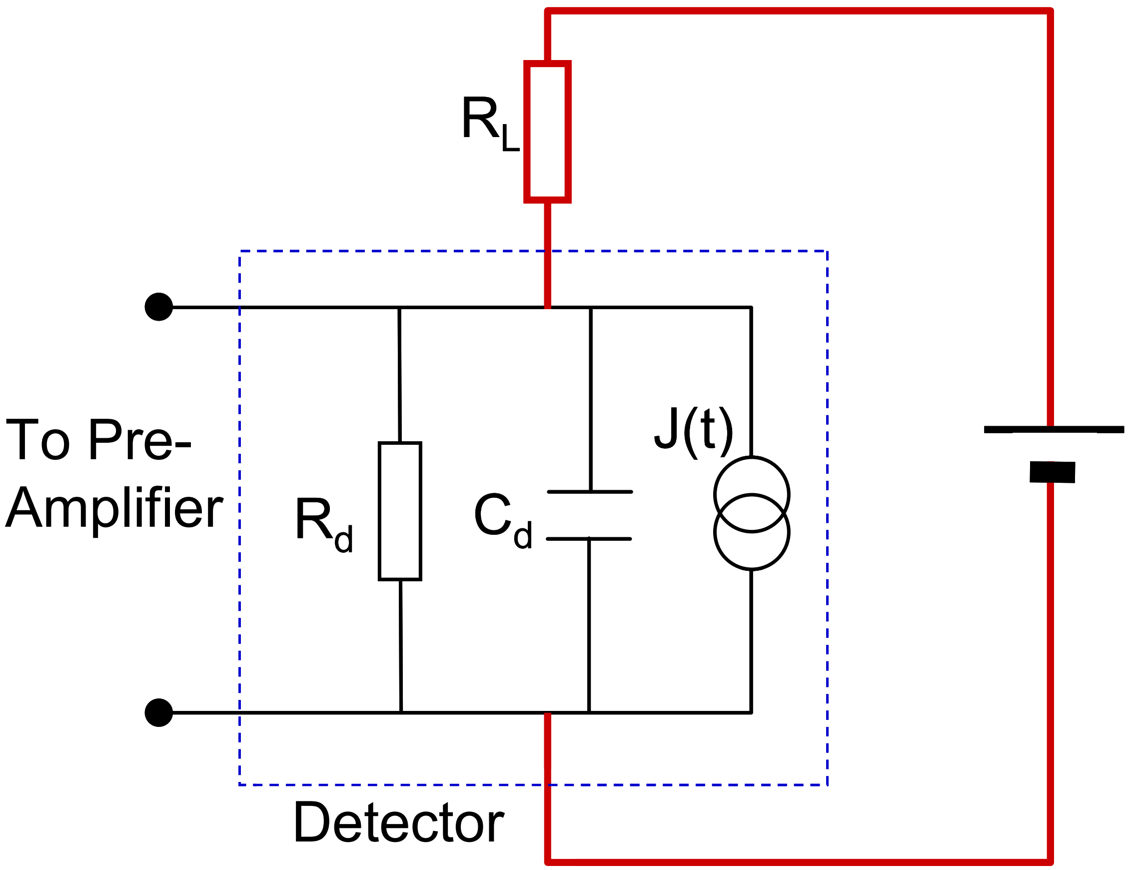


Figure 3.2 The equivalent circuit of the detector (inside the dashed box) and the circuit used for measurement of the photoresponse (C_d = detector capacitance, R_d = detector resistance).

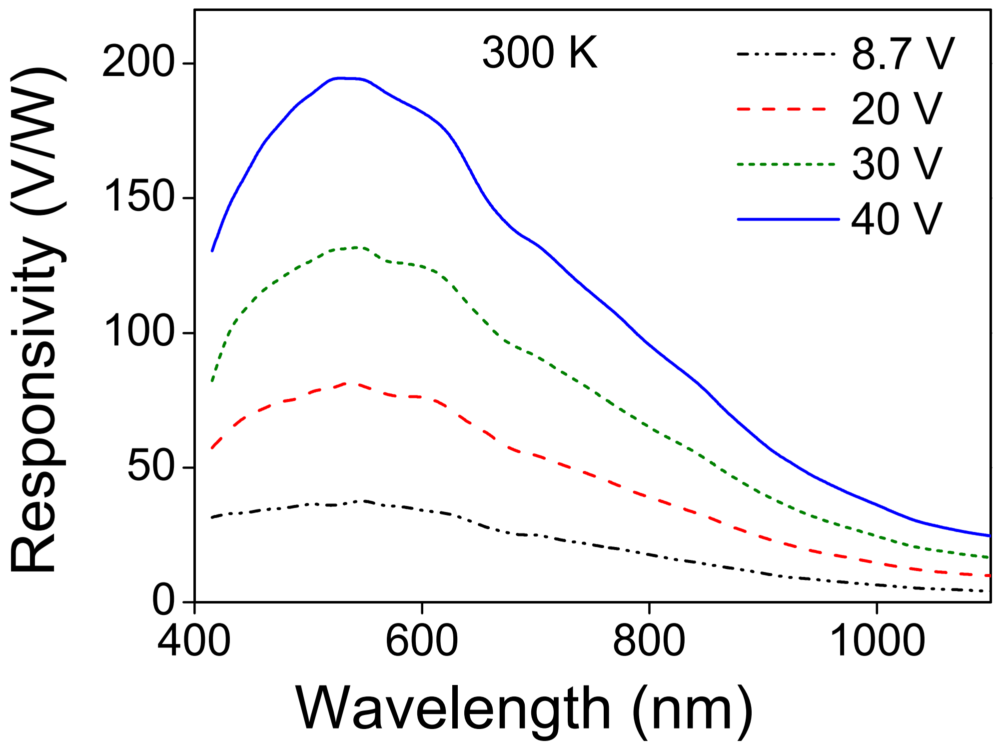


Figure 3.3 The responsivity of the detector under different bias voltages. (85 MΩ resistor was used as Load resistor R_L and chopping frequency was 57 Hz).

When the total amount of PbS incorporated into the wax reached ~15 % by weight, PbS begins aggregating masking the detection of this effect. Almost complete absorption of incident radiation avoiding any percolation threshold would be possible by increasing the film thickness. In general, near percolation thresholds, systems tend to be noisy[92] and therefore optimization should be achieved keeping N well below N_C and appropriate adjustment of *s.* In the present investigation no attempt was made to optimize the system. In the measurement reported, the bias voltage was varied from 1 to 40V with an 85 MΩ load resistor (R_L). The plot of voltage responsivity vs. the wavelength of the incident radiation is shown in Figure 3.3 and the responsivity at the peak absorption ($\lambda = 540$ nm) was found to be 195 V/W at a bias of 40V and the specific detectivity under the same condition was determined as 3×10^8 cm Hz$^{1/2}$/W.

3.4 Conclusion

The reported detectors based on displacement current of the CQD embedded dielectric media have the advantage of only detecting the change in the light intensity so that they are not saturated by the background at 300 K, while the inherent high resistance in the capacitor approach eliminates the dark current problems present in most room temperature detectors. The responsivity of the present system can be compared to that of a photoconductive detector of the same bulk material as follows. As the photoconductive current density is $J_c = en\mu E\,(\mu = e\tau/m)$, $\mu =$ mobility, τ = scattering time, m = carrier effective mass and taking low frequency molecular exciton polarizability = $\dfrac{e^2}{\varepsilon_0 m\omega_0^2}$ ($\omega_o =$ exciton binding energy / $\hbar$), we can be obtain $\dfrac{J}{J_C} = \dfrac{\omega s N A}{\tau\omega_0^2} \sim 10^{-5}$ ($A =$ detector area = 1 cm^2, ω = 57 Hz, $N = 10^7$ and exciton binding energy = 10 meV). Due to the less than ideal coverage of CQDs in the dielectric (i.e., $N = 10^7$) the responsivity is very low compared to a photoconductive detector. Owing to its low melting point and brittleness, paraffin wax is not the best material to embed CQDs. However, due to the simplicity of the preparation, paraffin wax was selected as the dielectric; any other dielectric material including silica, glass or polymers could also be used. Better methods of preparation of PbS CQDs and use of other embedding dielectrics should improve the performance, when N is increased by several orders of magnitude. It is straightforward to extend the concept proposed here to QDs of other materials and nanowires. Carbon nanotubes which possess high polarizabilities[93] with band gap taliorability would provide an option to fabricate multiband detectors. Thus more versatile photon detectors may be developed if nanotubes are used instead of quantum dots. An additional merit is the sensitivity to polarized radiation if the nanotubes are aligned in the dielectric medium. Furthermore, large aligning torques can be applied to carbon nanotubes, in a highly resistive dielectric medium compared to a conducting medium. The effects of multi-exciton production will also be reminiscent in these detectors, especially if PbS is replaced by PbSe; In PbSe-conducting polymer photon detectors, enhancements in quantum efficiency originating from impact ionization has been observed[94].

The optimization of the detectors based on this technique requires further studies on quantum dot/ nanotube impregnated dielectric media and assessment of noise and ways minimizing it. It is also important to test the room temperature operability using CQDs that absorb longer wavelengths. Displacement current measurements in CQD embedded dielectric media could also give useful information on properties of low dimensional semiconductors.

Chapter 4

Split-off transition based uncooled infrared detectors for 3-5 μm and beyond

4.1 Introduction

Uncooled infrared detectors have attracted much attention recently due to their numerous applications in data communication, fire fighting, electrical circuit trouble shooting, surveillance, bio-molecular identification and the fields of astronomy and defense. Avoiding cryogenics for detector cooling can reduce the cost and weight: simplifying the sensor systems and allowing for widespread usage. One of the major drawbacks of infrared photon detectors is the need for cooling far below room temperature in order to suppress the thermal generation processes. These thermally generated carriers (dark current) effectively reduce the signal to noise ratio of the detector. The cooling requirements introduce difficulties for low cost, compact lightweight, and logistics for most infrared systems. Numerous studies have been conducted over the years to find new material systems and architectures for uncooled IR photon detection[95,96]. An important intrinsic semiconductor material for uncooled infrared photodetector applications[95] has been HgCdTe, which recently showed[97] 200 K operation. Narrow bandwidth quantum well[98] and quantum dot based detectors[99] working at room temperatures have also been reported. Extended InGaAs p-i-n photodiodes responding up to 2.6 μm in ambient conditions are commercially available. One of the drawbacks associated with HgCdTe detectors is the lack of possibilities for optimization because of the problems caused by lattice, surface, and interface instabilities. These problems originate from weak bonding characteristics of II–VI semiconductors and from high Hg vapour pressure. Weak bonding reduces the strength of the material, resulting in poor mechanical properties and creating difficulties in material processing. Moreover high Hg vapour pressure makes the composition control over a large area difficult, causing serious problems for the focal plane array applications. This has intensified the search for alternative infrared material systems. As a well developed material system GaAs is free of most of the above problems.

Highly *p*-doped GaAs exhibits enhanced absorption around 2- 4 μm range, beyond the free carrier absorption, due to light hole (L-H) / heavy hole (H-H) to split-off band transitions. The split-off band effects have been experimentally observed in the emission of GaAs metal semiconductor field effect transistors[100] and have enhanced the response of GaInAsP[101] quantum wells and GaAs[102] quantum wells. Extensive theoretical studies on the importance of the spin split-off band and the tunnelling properties of the holes through $Al_xGa_{1-x}As$/GaAs heterostructures are reported elsewhere[103]. This study was based on two sets of p^+-GaAs/$Al_xGa_{1-x}As$ heterostructure detectors. The first split-off response was observed in a HEIWIP structure designed for the detection of MIR radiation. The second set of samples was designed to study the effects of the free carrier threshold wavelength on the split-off response.

4.2 Heterojunction Interfacial Workfunction Internal Photoemission (HEIWIP) detectors

The active region of the basic HEIWIP detector consists of one or more periods, each consisting of a doped emitter and an undoped barrier layer. These multiple emitter/barrier layers are sandwiched between two highly doped contact layers as shown in Figure 4.1(a). Depending on the doping required for ohmic contacts, the top contact may also serve as the top emitter layer. A *p*-type band diagram for a single period of a detector is shown in Figure 4.1(b). Here, the workfunction (Δ) is given by $\Delta = \Delta_d + \Delta_x$ where Δ_d and Δ_x are the contributions from doping and the Al fraction, respectively[104]. The dashed lines indicate the valence-band edge if the barriers were GaAs. As the Al fraction is reduced, Δ will be limited by Δ_d, which in turn is a homojunction detector[105,106]. The detection mechanism can be divided into three main processes**;** i) the photoabsorption which generates excited carriers, ii) escape of the carriers, and iii) collection of the escaped carriers. Free carrier absorption in the emitter layers initiates the process. The excited carriers then escape from the emitter layer by internal photoemission at the interface between the emitters and barriers. The applied electric field will sweep the carriers out of the active region. This standard mechanism can also occur for electrons in the conduction band of *n*-type detectors. By having a high enough doping to have the scattering length similar to the emitter thickness, the carriers will scatter before the wave function can interfere with itself and hence will not form discrete quantum states inside the well. This makes the carrier distribution in the emitter three dimensional but still bound. The HEIWIP detectors designed for the MIR and FIR ranges also showed response peaks at NIR region[22]. These peaks, which matched the GaAs split-off, were superimposed on the free carrier response.

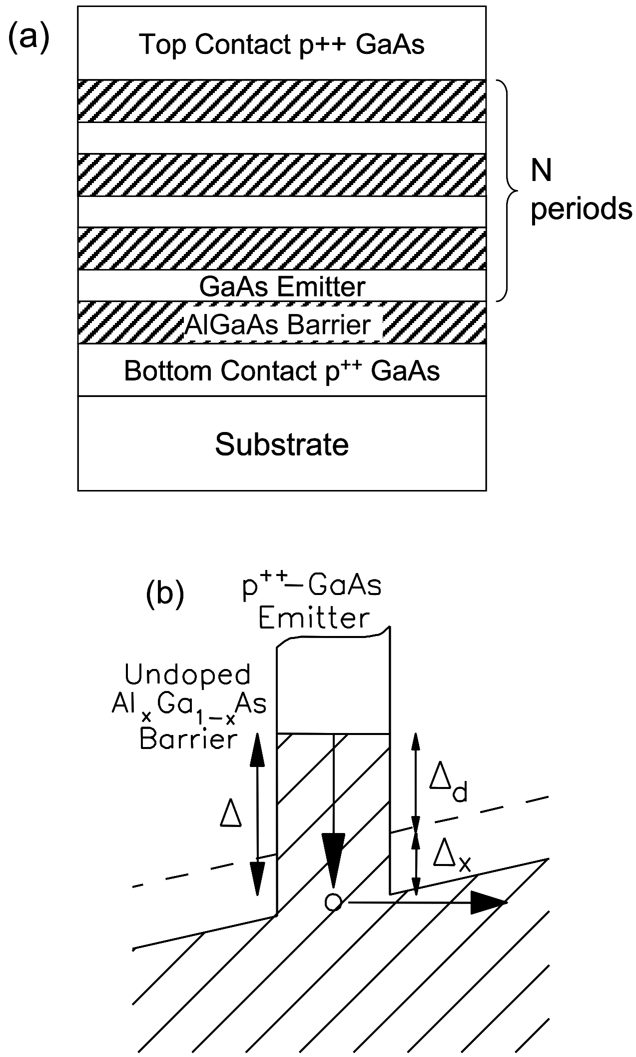


Figure 4.1 (a) Typical structure of a GaAs emitter AlGaAs barrier HEIWIP detector. (b) Band diagram showing the workfunction (Δ) for photoemission of carriers. Here, Δ is given by $\Delta = \Delta_d + \Delta_x$ where Δ_d and Δ_x are the contributions from the doping and the Al fraction, respectively. The dashed lines indicate the valence-band edge if the barriers were GaAs.

4.3 Split-off band detector mechanism

A split-off detector has a similar structure as HEIWIPs, described in Section 4.2. The emitters must be *p*-doped since the transitions involve are intra-valence band. A band diagram (E-k) for an emitter region of the detector is shown in Figure 4.2. In order to explain the detection mechanism, three valence bands will have to be considered as shown in Figure 4.2 , the L-H and H-H bands which are degenerate at k = 0, and the split-off band which is separated from them by an energy $E_{E\text{-}SO}$. Under equilibrium conditions, a *p*-doped region will have a Fermi level in the L-H and H-H bands, but above the split-off band maximum. The arrows in the Figure 4.2 indicate the possible threshold transition mechanisms. These transitions could be direct or indirect. Once the carrier is in the split-off band, it can escape directly or scatter back into the L-H/H-H bands and then escape. For a direct transition (shown by arrow 1 solid part) there is no phonon involved so k is conserved. The excited carrier then must scatter back to the L-H band (or possibly the H-H band although this involves much higher k changes) in order to escape. This phonon involved scattering is shown by the dashed arrow. The threshold for this process is determined by the difference of energy in the light and split-off hole bands at k corresponding to the Fermi energy shown by E_F in the Figure 4.2. For the indirect transition (shown by arrow 2) a phonon is involved and higher energies are possible. The high energy transition which is below the barrier in the split-off band could escape directly (it is not shown as this is a purely spatial change and does not affect the k-space picture). The indirect absorption has two thresholds. For escape with a scattering it is the difference of the Fermi energy and the split-off band at k=0 in the emitter ($\Delta_{E\text{-}SO}$). For escape without scattering, it is the difference between the Fermi energy and the split-off band at k=0 in the

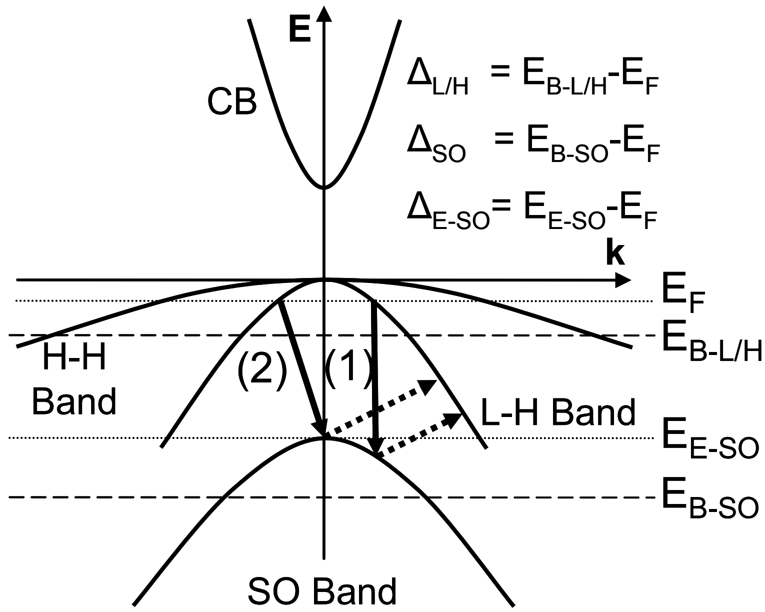


Figure 4.2 A band diagram for an emitter region of the detector, illustrating the different IR detection threshold mechanisms. The horizontal dashed lines $E_{B-L/H}$ and E_{B-SO} indicate the L-H/H-H and split-off band maximum (k=0) positions in the barrier. The horizontal dotted lines E_F and E_{E-SO} indicate the Fermi energy and the split-off energy in the emitter at k=0. The arrows indicate the possible threshold transition mechanisms (1) direct transition from L-H band to SO band followed by scatter back to L-H band (2) indirect transition followed by scatter back to L-H band.

barrier (Δ_{SO}). The horizontal dashed lines $E_{B-L/H}$ and E_{B-SO} indicate the L-H/H-H and split-off band maximum (at k=0) level in the $Al_xGa_{1-x}As$ barrier. The horizontal dotted lines E_F and E_{E-SO} indicate the Fermi energy and the split-off energy at k=0. Even though Δ_{SO} is constant for a given material system, $\Delta_{L/H}$ (also known as free carrier threshold) can be adjusted by varying Al fraction x and the doping concentration of the emitter layer[18]. The $\Delta_{L/H}$ is the lowest barrier for the excited carriers in L-H /H-H bands which determine threshold wavelength for the free carrier mechanism and also major contribution for dark current hence determined maximum operating temperature. By making $\Delta_{L/H}$ larger it is possible to make split-off transitions as dominant detection mechanism and increase operating temperature. However, higher $\Delta_{L/H}$ can lead to lowering the response by reducing the escape probability of impact ionizing carriers due to high-energy photons. The patent pending[20] detection mechanism described above involves free carrier transitions. This is different from the response observed previously from Si/SiGe QWIP detectors,[107] which used transitions from bound states to either a bound split-off band state, or a continuum state which is a mixture of the light, heavy and split-off hole bands.

4.4 Split-off response of HEIWIP detectors designed for MIR

The split-off response was first demonstrated using a detector designed for the 10-15 μm range with a 20 μm threshold[108], which was not designed for optimum split-off response. The detector labeled as HE0204 consisted of 16 periods of *p*-doped 188 Å GaAs emitters doped to 10^{18} cm^{-3} with carbon, and 1250 Å $Al_{0.12}Ga_{0.88}As$ barriers. The top and bottom contacts were 1×10^{19} cm^{-3} *p*-doped GaAs layers with 0.2 and 0.7 μm thicknesses, respectively. A schematic of a processed detector is shown in Figure 4.3. The measured responsivity in the 1.2-5 μm split-off range for HE0204 at 80-130 K is shown in Figure 4.4. A peak response of 0.55 mA/W was seen at 105 K at

2 μm. As the temperature was further increased, the response decreased, and was not measured beyond 130 K. The increase in response with temperature may be related to phonon effects on the escape rate for excited carriers. The total quantum efficiency determined by dividing the photocurrent by the incident photon rate was ~0.04, and at 1.8 μm a specific detectivity (D^*) of 2.6×10^4 Jones was obtain at 90K. This value is low since these detectors were designed for operation at much lower temperatures. The increased barrier in an optimized detector should reduce the dark current and hence improve D^*. The two steps seen in the response at 2.8 and 3.4 μm are probably caused by the thresholds for mechanisms (1) and (2) respectively (see section 4.3). This indicates that the use of high doping is the preferred approach. Based on previous experimental results and the standard thermionic current calculations, the dark current should not increase significantly, as doping is increased until the defect assisted tunneling dominates. If the doping is kept below these high values, the absorption is increased, and therefore the response and hence the background limited infrared performance (BLIP) temperature should increase.

In order to understand the response observed using the split-off band, calculations were carried out to determine the relative absorptions for the free carrier and split-off responses. The first step was to use a **k·p** model, similar to that used in quantum dots,[109] and quantum wells to calculate the L-H, H-H and split-off hole energy bands. The absorption coefficient was then calculated as a function of photon energy $\hbar\omega$ from the energy states in the band. The calculation was done for a 1 μm thick GaAs layer that was *p*-doped to 3×10^{18} cm^{-3}. The absorption by the split-off band was over an order of magnitude larger than for the free carrier absorption associated with the transition between L-H/H-H bands as shown in Figure 4.5, indicating the relative improvement of the split-off mechanism in this range.

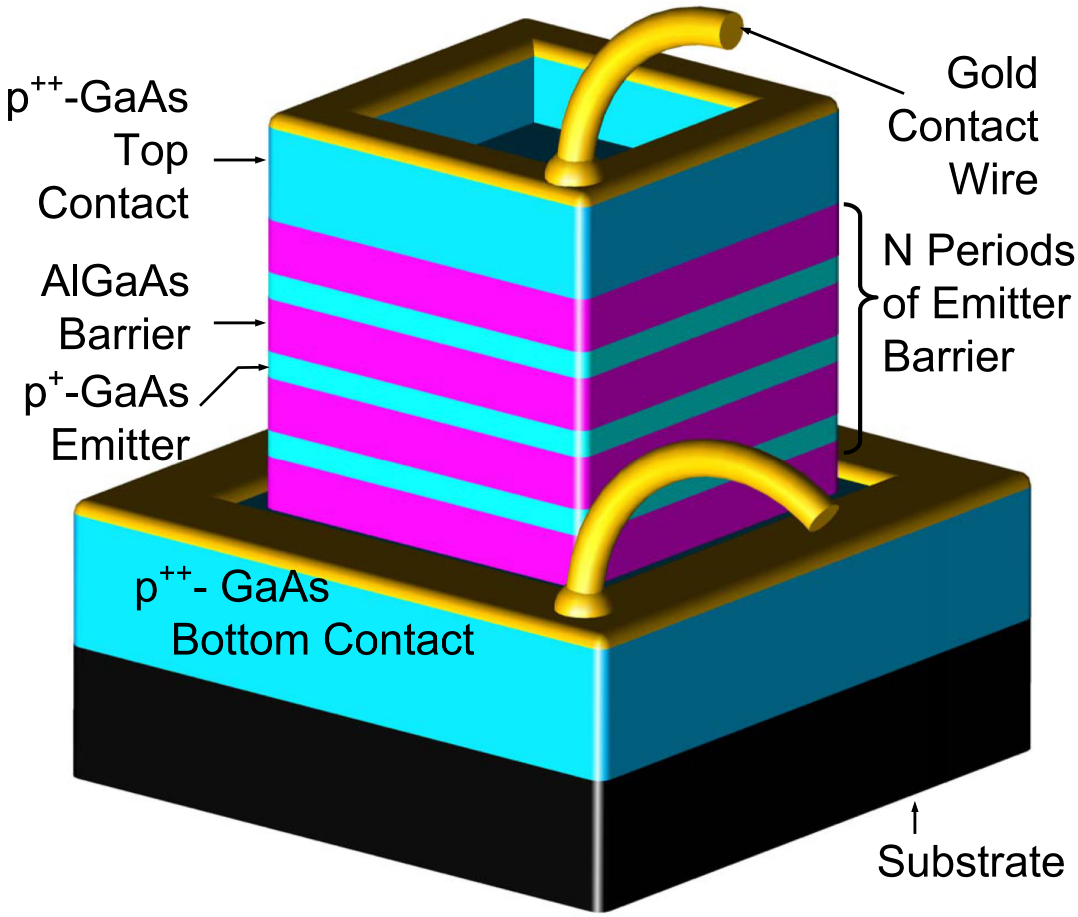


Figure 4.3 A schematic of a mesa of the processed detector showing the window etched in the top contact for front-side illumination. The number of emitter barrier periods, emitter and barrier thickness, and doping values were different for each design. Part of the bonded contact wires are also shown here.

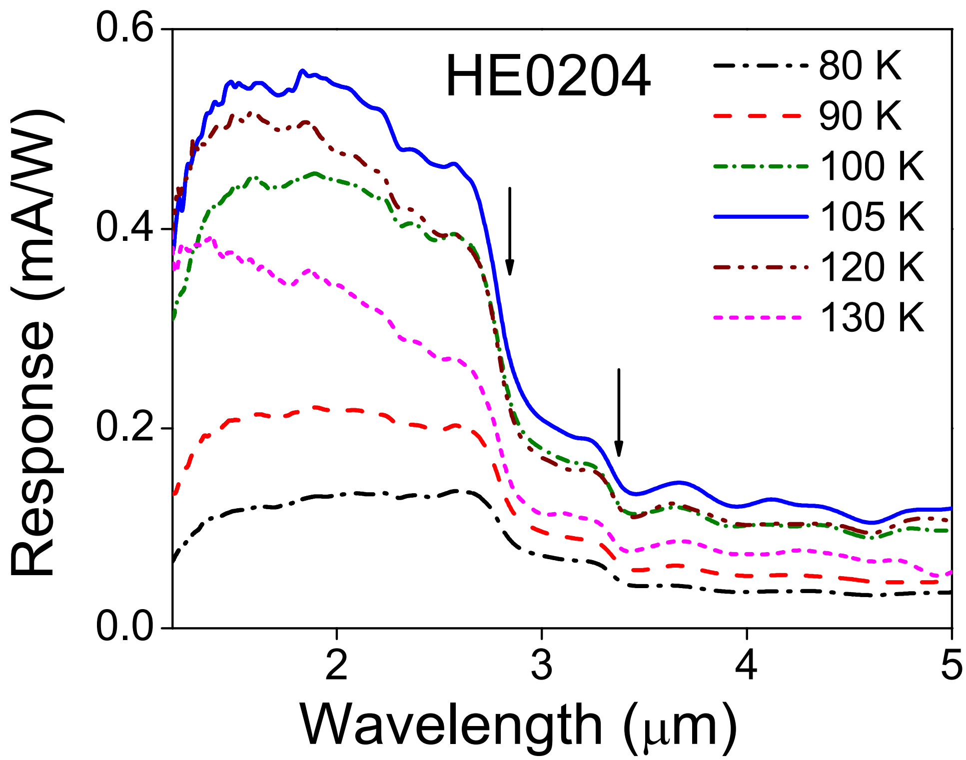


Figure 4.4 Measured responsivity of HE0204 at temperatures from 80 K to 130 K under 5 kV/cm bias. The responsivity increased up to 105 K and then started to decrease. The peak response was 0.55 mA/W at ~2.0 μm. The two steps indicated by arrows in the response at 2.8 μm and 3.4 μm are due to the thresholds for mechanisms (1) and (2), respectively.

As shown in Figure 4.6 the measured absorption increased in the split-off region compared to the expected free carrier absorption. The step seen in response at 3.4 μm is in good agreement with the calculated results shown in Figure 4.5. However, there is a small discrepancy with the drop at 2.9 μm in the calculated curve. The experimental step is ~ 0.15 μm wide, (possibly due to the effects of the photoemission process) and the threshold may be longer than it appears in Figure 4.4. The split-off response observed in a HEIWIP detector designed for threshold wavelength of ~20 μm showed a maximum operating temperature of 130 K. By reducing the free carrier threshold to ~5 μm, uncooled operation should be achieved with a D* of ~5×10^6 Jones. Based on above results, 3 sets of split-off band detectors were designed, grown and measured results are discussed in the following section.

4.5 Uncooled split-off detector design and experimental results

The detector structures SP1, SP2, and SP3 with three different $\Delta_{L/H}$ values of 155, 207, 310 meV (the corresponding Al fractions for those $\Delta_{L/H}$ values are x = 0.28, 0.37, 0.57 giving threshold wavelengths of 8 μm, 6 μm, and 4 μm, respectively) were grown on semi- insulating GaAs substrates with a 0.7 μm thick, 1×10^{19}cm^{-3} *p*-doped GaAs bottom contact layer, of GaAs followed by 30 periods of a 600 Å, undoped $Al_xGa_{1-x}As$ barrier, and a 3×10^{18}cm^{-3} *p*-doped, 188 Å GaAs emitter. The top most emitter was 0.2 μm thick and *p*-doped to 1×10^{19}cm^{-3} in order to use as the top contact layer. The detectors were processed by wet etching to form square mesas with sides 400, 600, 800, and 1000 μm. Ti/Pt/Au ohmic contacts were evaporated onto the top and bottom contact layers. A ring contact was used on the top surface and a window was opened through the top contact for front side illumination.

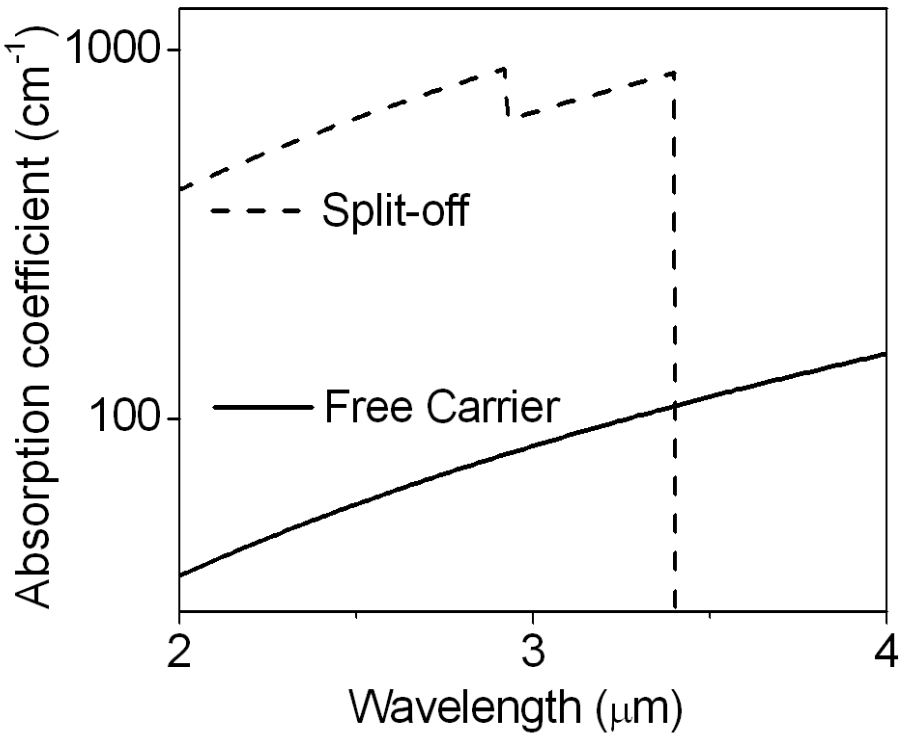


Figure 4.5 The calculated absorption coefficient for the free carrier and the SO band mechanisms in a 3×10^{18} cm^{-3} *p*-doped GaAs layer.

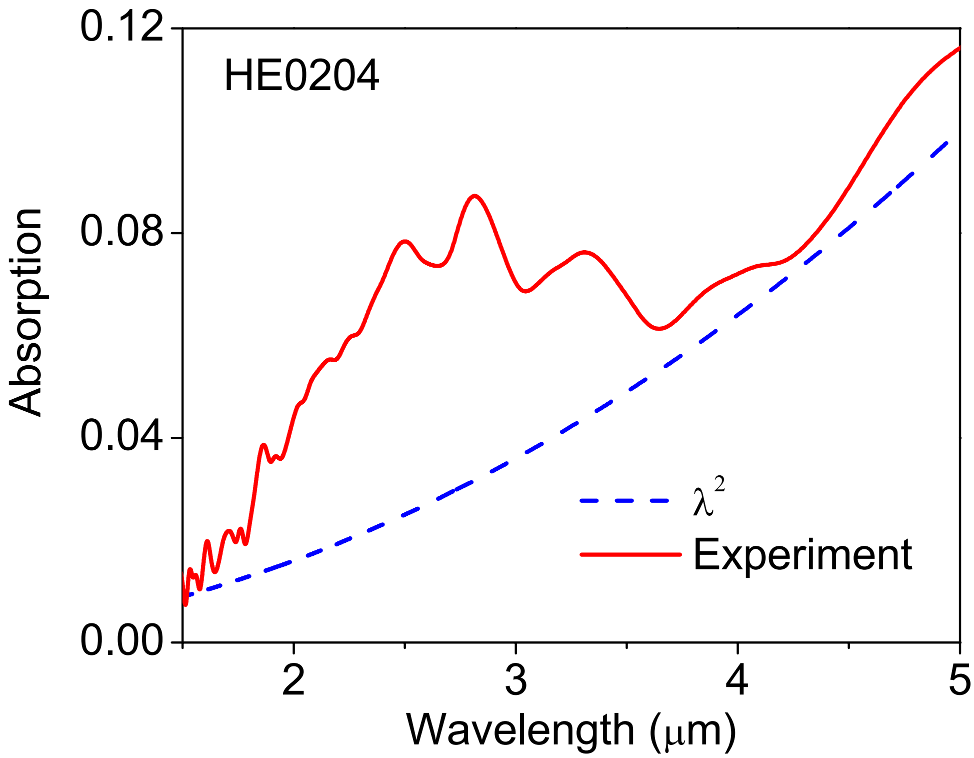


Figure 4.6 Measured absorption for the sample HE0204 showing an increased absorption from the split-off transitions. The dashed line shows the expected free carrier response varying as a λ^2.

The sample pieces with ~4 mm × 8 mm dimensions and 10 -12 different sized mesas were mounted on standard chip carriers, and gold wires were wire bonded from the top and bottom electrical connections. A schematic of a single mesa of the detector is shown in Figure 4.3. The current voltage (I-V) characteristics were measured at different temperatures from 70 to 300 K. As shown in Figure 4.7, the dark current density at 1V bias reached the same order at 140, 190 and 300 K for samples SP1, SP2, and SP3, respectively. Comparison of the measured dynamic resistance and the dark current density for the 3 samples are presented in Table 4.1. The spectral response of the detectors SP1, SP2, and SP3 were measured using a Fourier transform infrared spectrometer (Perkin-Elmer system2000) at temperatures up to 140 K, 190 K, and 300 K, respectively. Figure 4.8 shows responsivity calibrated using a bolometer for each detector under 4 V bias. These responsivities were also verified with a calibrated InGaAs photodiode at 2.5 μm. Figure 4.9 shows the responsivity of the detector SP3 at 300 K under four different biases with the maximum responsivity of 0.29 mA/W at 300 K and 2.5 μm. The noise current density was measured using a dynamic signal analyzer (Stanford Research SR785) with the sample mounted in an optically and electrically shielded dewar. Batteries were used to bias the sample to minimize noise components from the bias source. The majority of noise is believed to be generation recombination and Johnson noise. Normalized detectivity (D*) was obtained (see Appendix A.4 for details) using the formula $D^* = (R \times \sqrt{A})/I_N$, where R is the responsivity (in A/W), A is the optically active area of the detector (in cm^2) and I_N is the noise current density (in $AHz^{-1/2}$). The calculated D* values for each sample at the optimum operating temperature is listed in Table 4.1. The response observed in the sample SP3 at room temperature appears to be due to a single emitter without the contribution of other emitters. Modifications of the design to activate more emitters would lead to increased gain hence improved response.

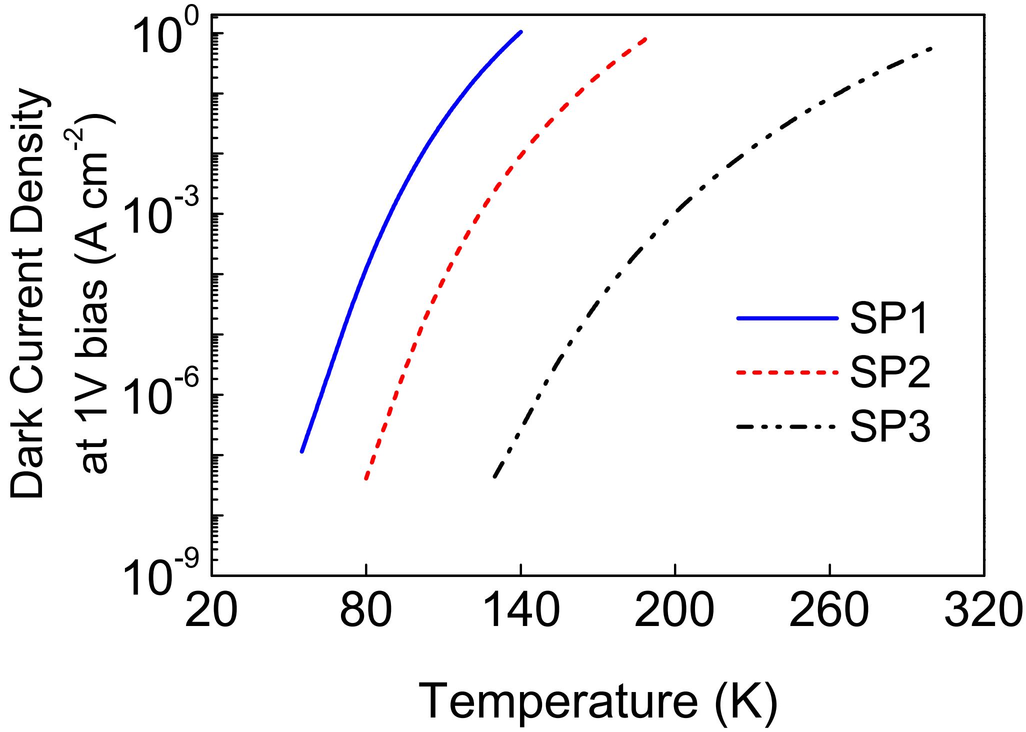


Figure 4.7 The dark current density vs. temperature for samples SP1, SP2 and SP3 under 1V applied bias. The samples SP1, SP2, and SP3 have barriers with Aluminum fractions 0.28, 0.37, and 0.57, respectively. The dark current densities reached ~1 A/cm^2 for each sample at 140, 190, and 300 K. Those temperatures were properly matched with the experimentally observed, optimum operating temperatures for each sample.

Table 4.1 Sample parameters at optimum operating temperatures (T_{max}). The dynamic resistance (R_{Dyn}), darkcurrent density (I_{Dark}) at 1V bias, and D* were experimentally measured. $\Delta_{L/H}$ is the designed band offset.

Sample	**$\Delta_{L/H}$ (eV)**	**T_{max} (K)**	**R_{Dyn} (Ω)**	**I_{Dark} (Acm^{-2})**	**D* (Jones)**
SP1	0.155	140	787	0.663	2.1×10^{6}
SP2	0.207	190	913	0.875	1.8×10^{6}
SP3	0.310	300	1138	0.563	6.8×10^{5}

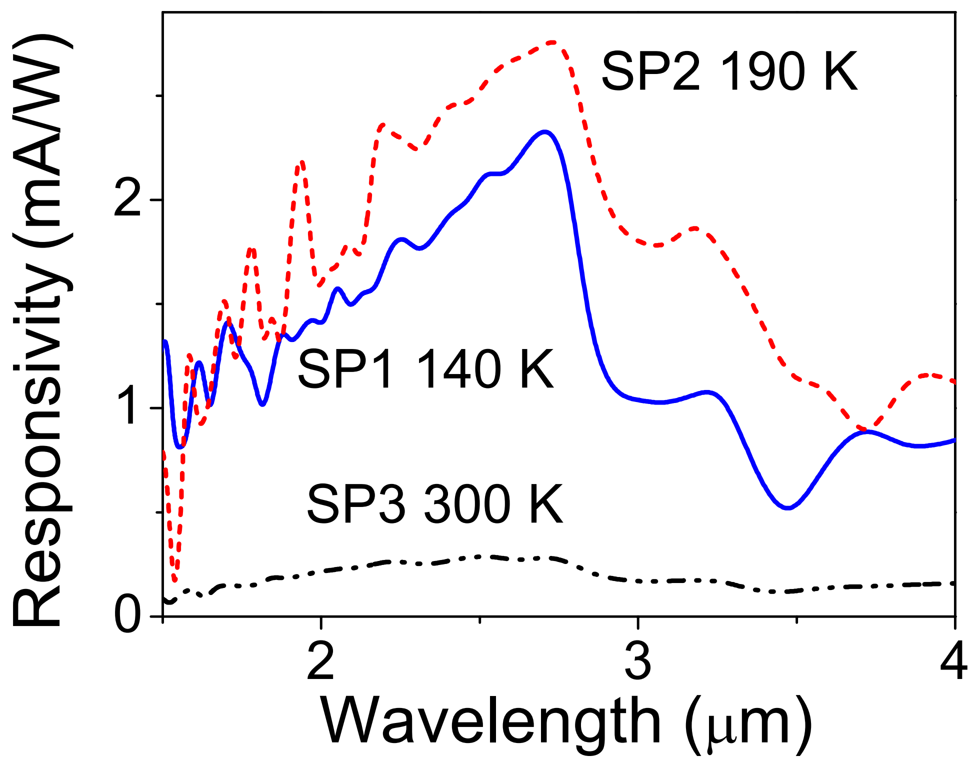


Figure 4.8 The responsivity of the samples SP1, SP2 and SP3 at the optimum operating temperatures of 140 K, 150 K and 300 K, showing peaks around 2.6 μm. The applied bias was 4 V for all three measurements.

As shown in Figure 4.9, the responsivity increases with the bias up to 4 V, but increased dark current (low dynamic resistance) reduced the response above 4 V bias. Even though the reported curves are at 300 K, responses have also been observed up to 330 K by heating the sample stage using a built-in heater. At 330 K, dynamic resistance was reduced to several hundreds of ohms and the measured signal deteriorates almost to the noise level. The threshold wavelengths for the response mechanisms 1 and 2 shown in Figure 4.2 can be identified in Figure 4.9 at 2.9 μm and 3.4 μm, respectively. At 300 K, mechanism (1) becomes dominant, as can be seen by the much larger step at 2.9 μm. The threshold for the free carrier response increased slowly with temperature due to increased number of carriers above the Fermi energy, which can give response at longer wavelengths. The SP1 and SP2 detectors exhibit optimum operating temperatures at 140 K and 190 K, respectively. The higher responsivity and D* seen for longer wavelength threshold samples are possibly due to impact ionization (gain), compared to the shorter threshold samples as shown in Figure 4.10.

4.6 Long wavelength (MIR and FIR) response observed in the split-off detectors

In addition to the expected split-off response in 2-4 μm range, a long wavelength response was also observed for the detector SP3. The responsivity in the 2-20 μm range for various bias voltages at a temperature of 250 K and 330 K are shown in Figure 4.11 (a) and (b), respectively. This response extends to much longer wavelengths than the free carrier response threshold of 4 μm. When the incident radiation was blocked, no signal was observed. The voltage response (V/W) increased for lower operating temperature as seen in Figure 4.11(a) and (b).

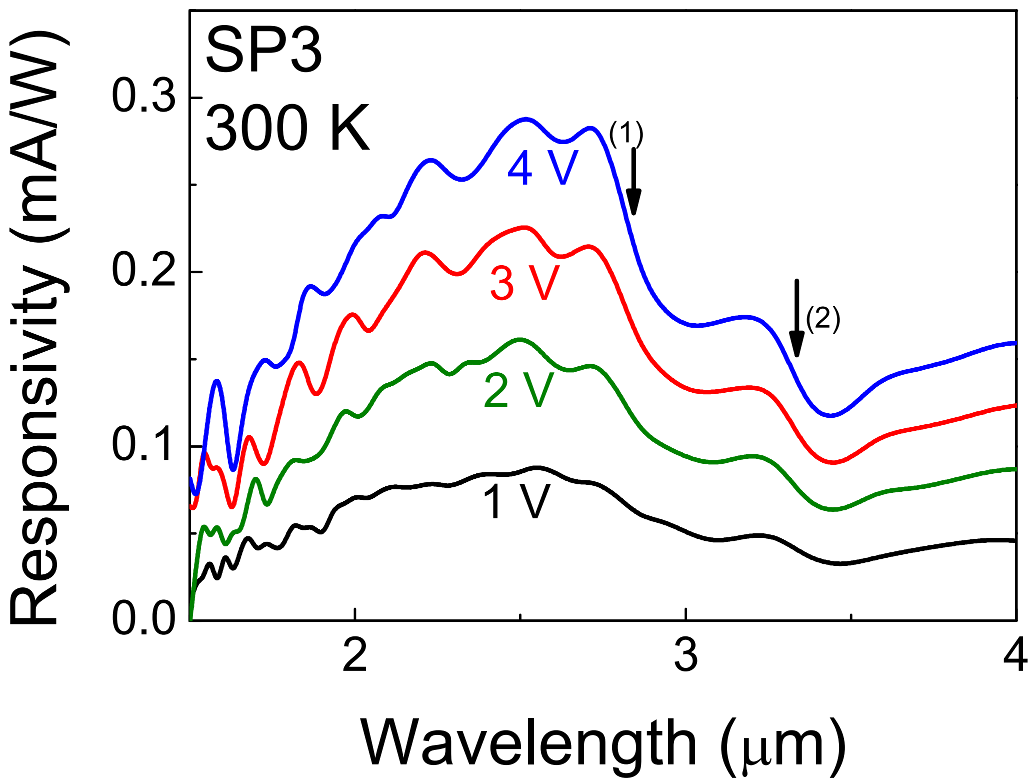


Figure 4.9 The measured responsivity of the sample SP3 under 4 different biases at 300 K showing peaks around 2.5 μm. The arrows 1 and 2 indicate the thresholds at 2.9 μm and 3.4 μm corresponding to the transitions 1 and 2 shown in Figure 4.2. The inset shows the responsivity of the samples SP1 and SP2 at 140 K and 150 K, respectively.

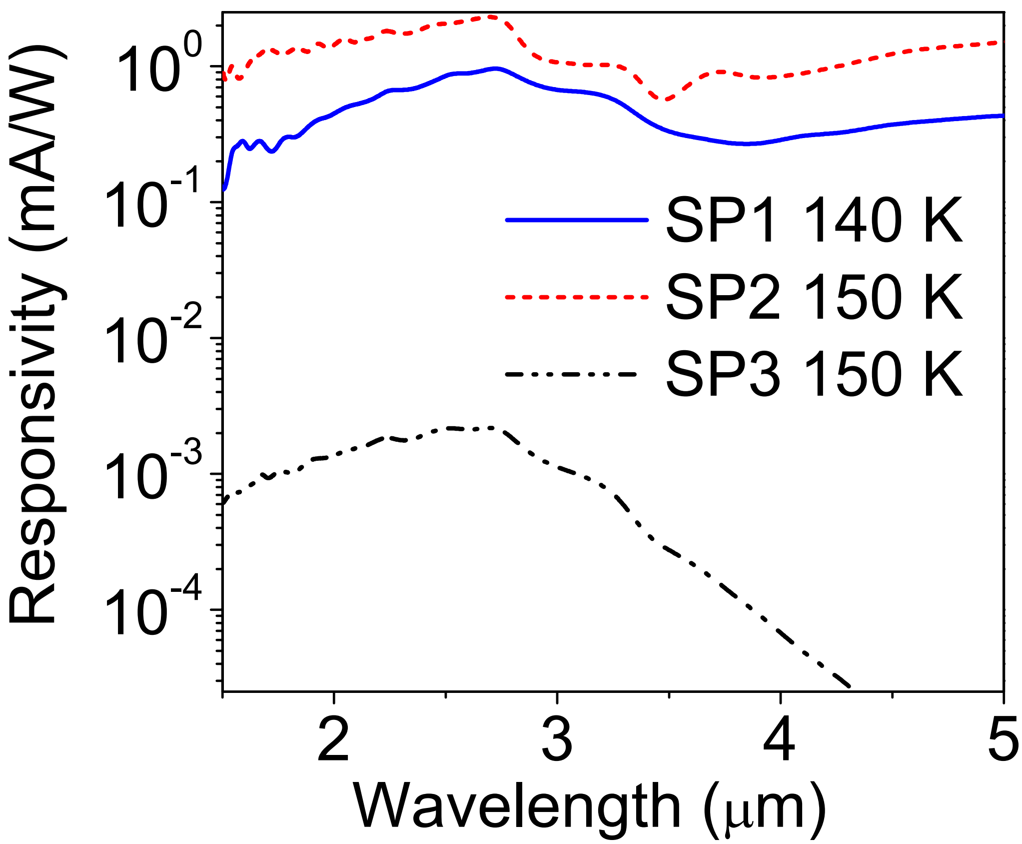


Figure 4.10 The responsivity of the samples SP1, SP2 and SP3 at 140 K, 150 K and 150 K, respectively. The higher responsivities seen for the longer wavelength threshold samples (SP1 and SP2) are possibly due to impact ionization (gain), compared to the shorter threshold sample (SP3).

The response is nearly flat above 4 μm, with small oscillations superimposed on it. The source of these oscillations is unknown at this time. The ratio of the long wave response (R_{lw}) to split-off response (R_{sp}) was increased with the temperature (R_{lw}/R_{sp} = 0.4 at 250 K and R_{lw}/R_{sp} = 1.1 at 330 K) showing that long wavelength response mechanism is dominant at higher temperatures. This indicates the response is probably associated with changes in the thermal population of the carriers at higher temperatures. A possible explanation is that the response originates with the excited carriers in the split-off band, which is 340 meV above the light/heavy hole bands. Unlike the standard semiconductor bolometric response in which the signal is due to the increased number of electrons and holes produced through thermal generation, the signal in this detector comes from an increase in the number of carriers in the split-off band at higher temperatures. The carriers in the split-off band could then scatter into the light/heavy hole bands above the barrier and escape. The escaped carriers would then be collected by an applied electric field, contributing to the photocurrent. In this approach, the observed photocurrent would be due to heating of the sample with the corresponding increase in the current as in a bolometer, but with a different mechanism (using the split-off band) for the change in the resistance producing the signal.

This SP3 detector was not optimized for thermal response, hence the efficiency could be further increased by making necessary modifications. The detector sample consisted of a 400 μm × 400 μm square mesa on a ~3 mm × 5 mm substrate with ~500 μm thickness. This had a large thermal mass reducing the temperature change and hence lowering the response as well as the response-speed. In addition, only a ~1 mm diameter area was illuminated and the device was not placed in an integrating cavity, as the bolometer, which greatly helps to increase the radiation incident onto the sample. By optimizing the sample size and placing it in an integrating cavity, it should be possible to improve the detector response. For a bolometer, the response is given by[110]

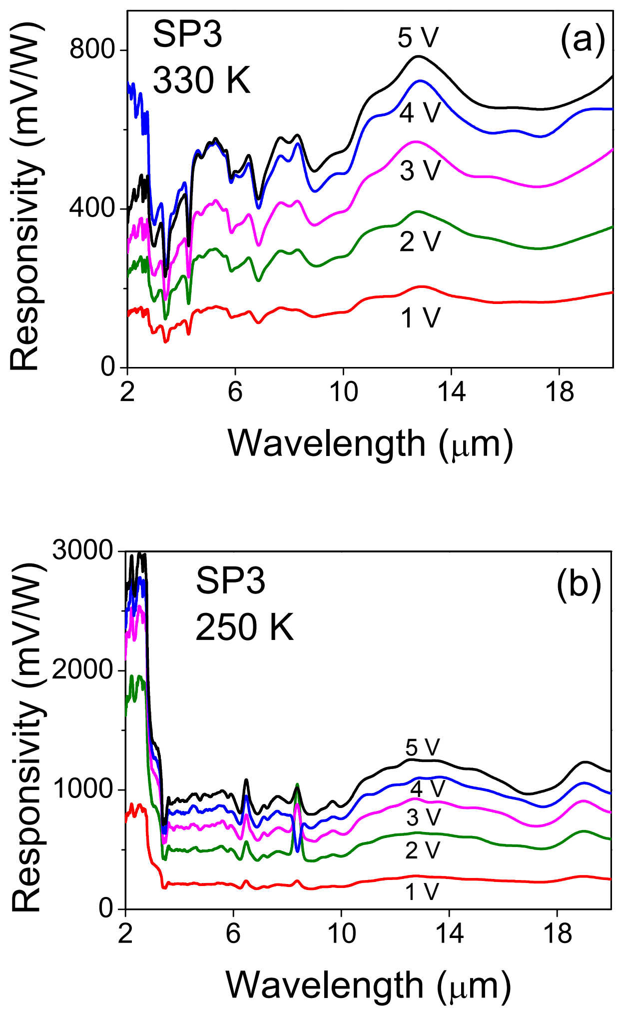


Figure 4.11 The responsivity of the split-off detector SP3 in the 2 to 20 μm range for different bias values in the 1- 5 V range. (a) At 330 K temperature, the maximum long wavelength response was ~800 mV/W for a bias of 5 V. Peak responsivity of the split-off range was ~475 mV/W at same bias. (b) At 250 K temperature the maximum long wavelength response was ~1250 mV/W and split-off responsivity was 3000 mV/W for a bias of 5 V. The response was nearly constant over most of the range.

$$R = \frac{IR_d \alpha}{G(R_d + R_l)} \tag{4.1}$$

where I is the current, R_d and R_l are the detector resistance and the load resistor, G is the heat flow rate, and $\alpha = (dR_d/dt)/R_d$ is the fractional rate of change in the detector resistance. For the split-off bolometer $\alpha = \Delta/(k_b T^2)$ where Δ is the split-off energy, T is the temperature, and k_b is Boltzmann's constant. The response observed for the detector corresponds to a heat flow rate of ~20 mW/K. For a small sample size, specifically mounted for use as a themal detector, the heat flow rate can be reduced to ~15 μW/K. Using this value, if the size of the sample is assumed to be reduced so that it does not limit the response, and the sample is assumed to be in an integrating cavity so that all the radiation is absorbed, the response of the detector should be increased to ~2000 V/W.

4.7 Future Work

4.7.1 Proposed materials for split-off threshold extension

The larger doping used to obtain a high absorption in the GaAs/AlGaAs based split-off detectors has shifted the peak of the split-off absorption to below 3 μm for process (1) which is the dominant response mechanism at higher temperatures. By reducing the doping, the split-off response peak can be moved to near 3.5 μm but the response will be reduced. Hence, the use of alternate materials would be the best option. Several possible semiconductor materials, their split-off energies, and the corresponding threshold-wavelengths are listed in Table 4.2. Arsenide and phosphide would be the best materials for 3-5 and 8-14 atmospheric windows since their split-off energies fall in this range. A bandgap vs. lattice constant plot of selected arsenide and phosphide materials is shown in Figure 4.12.

Table 4.2 The Split-off band offset energy for different materials. Nitride and phosphide compounds are of interest for wavelengths beyond 10 μm. The split-off energy of the quaternary alloy InGaAsP can be varied from 110 - 379 meV by changing the As fraction.

Material	**Δ_{SO} (meV)**	**λ_{SO} (μm)**
InP	110	11
GaP	80	16
AlP	70	18
InAs	410	3.2
GaAs	340	3.6
AlAs	300	4.1
InN	3	410
GaN	20	62
AlN	19	65
$In_{1-x}Ga_xAs_yP_{1-y}$	110 - 379 $(0.11+0.421y-0.152y^2)$	3.3 - 11
$In_{1-x}Ga_xP$	93 - 101 $0.101+0.042x-0.05x^2$	12.3 - 13.3

By using GaAsP as the emitter, the split-off threshold (λ_s) could be tailored. An emitter with $GaAs_{0.4}P_{0.6}$ should give a split-off threshold near 5 μm, allowing full coverage of the 3-5 μm range. However, there would be significant strain between the emitters and barriers in a structure that only used GaAsP emitters and AlGaAs barriers. The strain associated with the high P fraction would introduce defects, reducing the material quality. As shown in Figure 4.12, well matched lattice constants of GaP and AlP (5.4505 and 5.4510 Å[111]) will allow an $Al_xGa_{1-x}P$ ternary alloy and GaP to be latticed matched for all x values from 0 to 1. Heterojunction devices with a *p*-GaP emitter and an $Al_xGa_{1-x}P$ barrier on a GaP substrate should behave similarly to the well known *p*-GaAs/AlGaAs system. The free carrier threshold of the *p*-GaP/$Al_xGa_{1-x}P$ system can be tuned to GaP split-off energy of 80 meV (16 μm) by changing Al fraction x but still keeping lattice matched conditions. Growth of these structures should not be a difficult task since characteristics of GaP/AlGaP heterojunctions,[112-114] superlattice,[115] heterostructure based detectors,[116] and solar cells[117] have already been reported. Processing of these devices will also be straight forward since different etching methods for GaP/AlGaP have also been developed.[118-120]

The $In_{1-x}Ga_xAs_yP_{1-y}$ lattice match to InP has been well studied[121-123] and widely used as QWIP[124-127], transistor[128], LED, and laser[129,130] structures. The split-off energy of $In_{1-x}Ga_xAs_yP_{1-y}$ can be tuned for a wide energy range from 0.11 to 0.379 eV (3.3 – 11 μm) by changing the As alloy fraction y from 0 to 1 ($E_{SO} = 0.11 + 0.24y$)[123]. Hence, a detector with *p*-InGaAsP emitter and InP barrier would be a potential choice for a tunable split-off threshold from ~3 –11 μm. Since the valence band discontinuity of the $In_{1-x}Ga_xAs_yP_{1-y}$/InP heterostructure can be expressed as[111] $\Delta E_V = 0.502y - 0.152y^2$, the free carrier threshold wavelength will also be tunable. However, in order to achieve similar or ~10% less ΔE_V compared to the E_{SO}, "y" should be with in 0.5 - 0.7 range. This corresponds to a threshold wavelength in the 4-5 μm range.

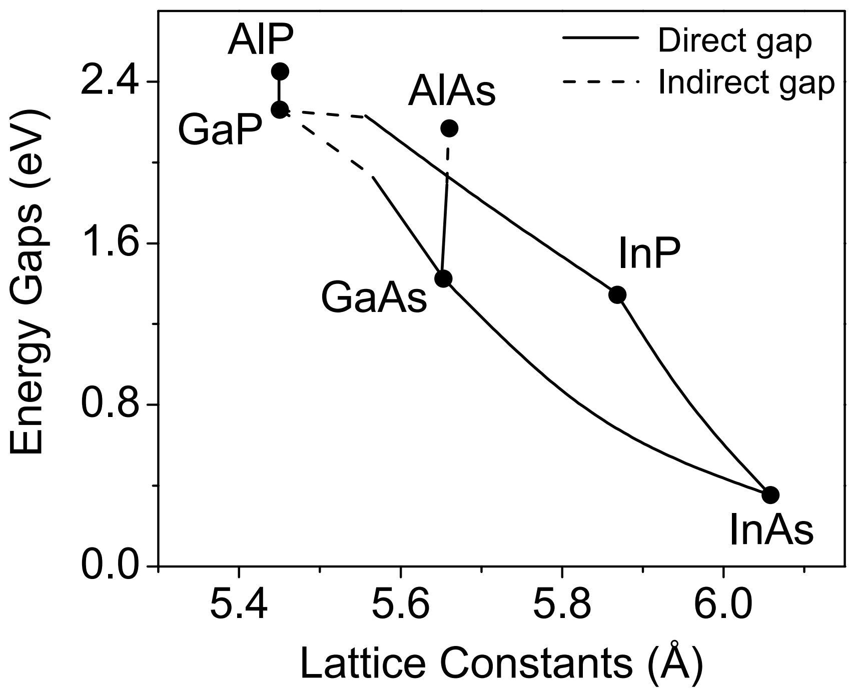


Figure 4.12 Band gap and lattice constant of selected III-V semiconductors (data were taken from "Handbook series on semiconductor parameters[111]".) Solid lines represent direct band region and dashed lines represent indirect band region. GaP/AlGaP would be a high-quality, lattice matched system similar to GaAs/AlGaAs. Also, InGaAs and InGaP can be lattice matched to InP and GaAs, respectively.

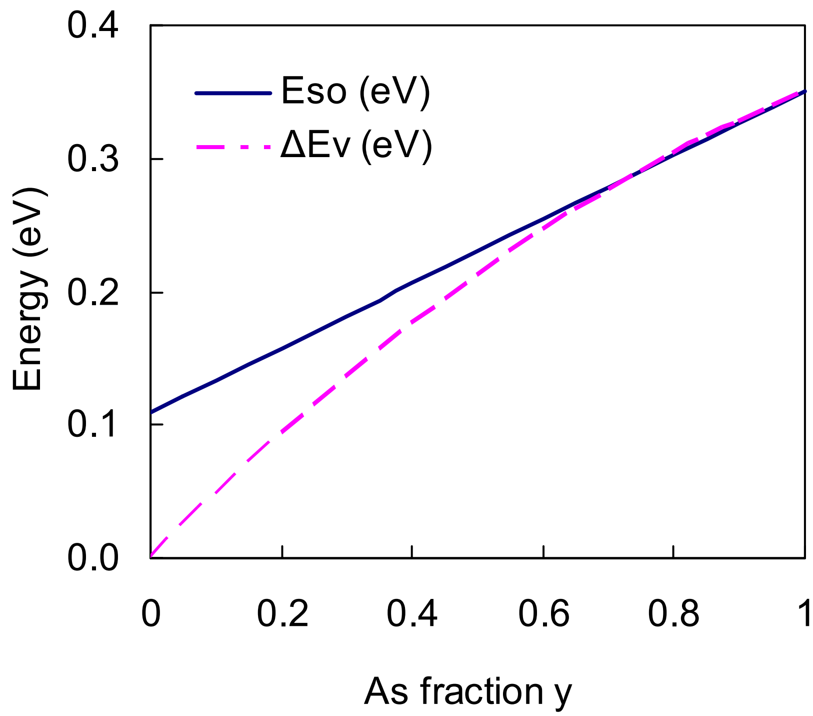


Figure 4.13 The variation of the valence band offset energy (ΔE_V) and split-off energy (E_{SO}) of the $In_{1-x}Ga_xAs_yP_{1-y}$/InP heterostructure with Arsenic fraction y. To achieve a similar or 10% less ΔE_V, the Arsenic fraction should be in the 0.5 - 0.7 range.

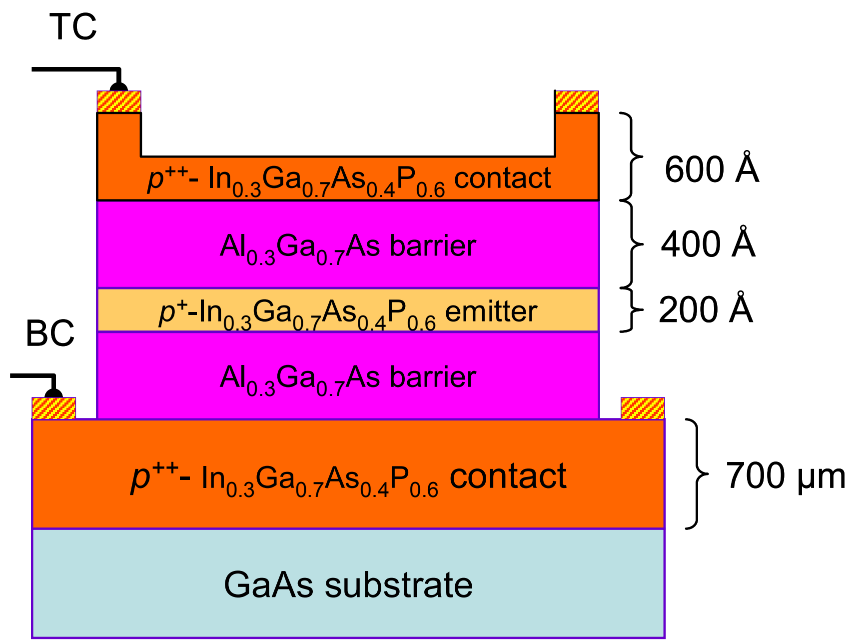


Figure 4.14 Schematics of the $In_{1-x}Ga_xAs_yP_{1-y}$ emitter and $Al_{0.3}Ga_{0.7}As$ barrier based detector structure, which is lattice matched to the GaAs substrate.

A calculated ΔE_V and E_{SO} for different Arsenic fractions y is shown in Figure 4.13. In order to keep lattice matched condition with InP, the $x \approx 0.47\ y$ condition should be satisfied. It is possible to grow InGaAsP with the In and P fractions chosen so that structure would be lattice matched to GaAs[131-133]. This would allow the freedom in tailoring the barrier by the use of AlGaAs. A device with $In_{0.3}Ga_{0.7}As_{0.4}P_{0.6}$ would be lattice matched to GaAs and would have a split-off threshold of 5.1 μm for process (1), which is the dominant split-off mechanism (see Figure 4.2). The barriers in an optimized detector are expected to be $Al_{0.30}Ga_{0.70}As$ to reduce the dark current. Thus the device structure would consist of 3×10^{18} cm^{-3} doped, 200 Å $In_{0.3}Ga_{0.7}As_{0.4}P_{0.6}$ emitters and 400 Å $Al_{0.30}Ga_{0.70}As$ barriers. A schematic digram of the detector structure is shown in Figure 4.14.

4.7.2 Dual-band detector design for the 3-5 and 8-14 μm atmospheric windows

A dual-band split-off detector is proposed for the practically important atmospheric windows in the 3- 5 μm and 8- 14 μm ranges. This structure is based on the well studied[134-142], lattice matched, GaAs/$In_{1-x}Ga_xP$/AlGaAs material system. $In_{1-x}Ga_xP$/AlGaAs/GaAs heterojunction based transistors[143-146], diode lasers[147,148], and tunnel diode based solar cells[149-151] have been reported. The lattice constant and band gap of $In_{1-x}Ga_xP$ can be expressed as 5.8687-0.4182x and 1.34 + 0.69x + $0.48x^2$ (for x < 0.63), respectively (see page 38 of Ref [111]). $In_{0.49}Ga_{0.51}P$ would be lattice matched to GaAs with a band gap E_g of 1.81 eV and spin-orbital splitting Δ_{SO} of 0.109 eV (11.3 μm). The idea is to use the p-GaAs/AlGaAs emitter barrier system to detect the 2 - 5 μm range and to use the p-$In_{0.49}Ga_{0.51}P$/AlGaAs emitter barrier system to detect the 8-14 μm range. A schematic of the detector structure is shown in Figure 4.15, and the band diagram under a biased condition is shown in Figure 4.16. This three-contact device architecture was previously reported[152] for a GaN/AlGaN, UV-IR dualband system, and it allows simultaneous identification of each wavelength band.

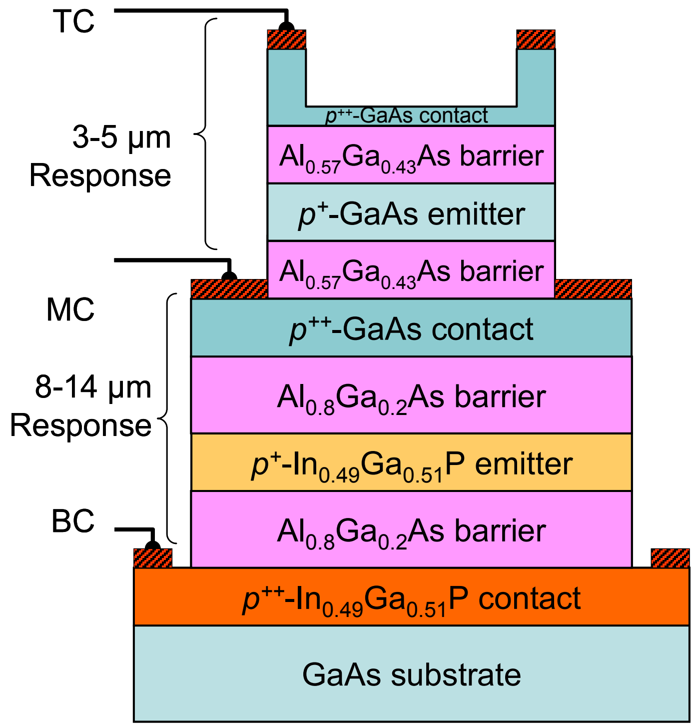


Figure 4.15 Proposed dual-band detector structure for the 3-5 and 8-14 μm atmospheric windows. A three-contact design will allow the separate detection of 3-5 μm radiation from the top to middle contact (TC – MC) and 8-14 μm range radiation from the middle to bottom contact (MC – BC). The Al fractions of the barrier (0.57 and 0.8) were selected so that the barrier height is equal to the split-off energy of the emitter layer.

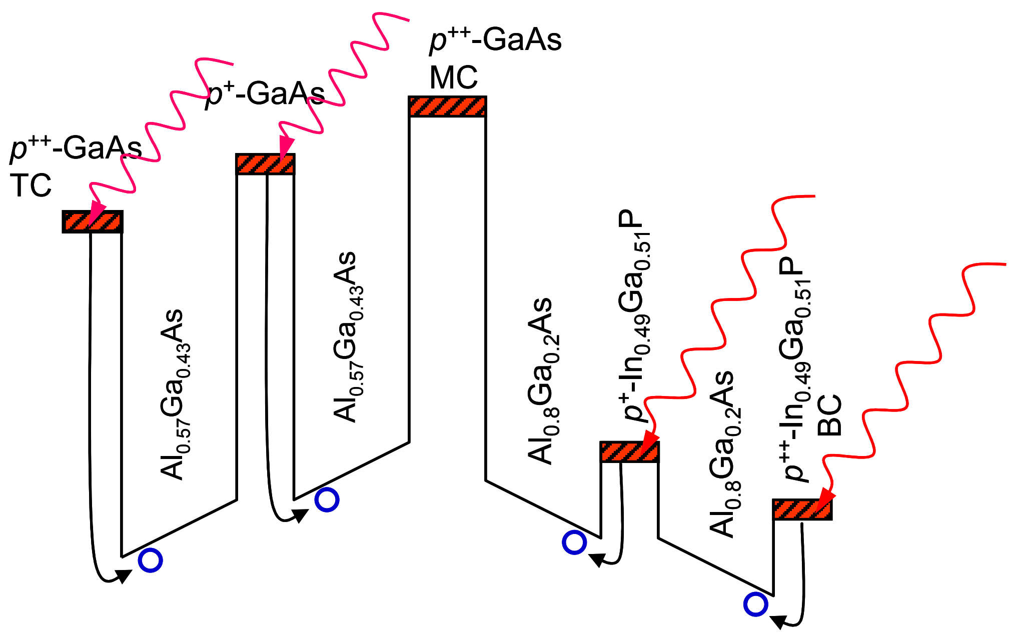


Figure 4.16 The band diagram of the proposed dual-band detector under a biased condition. The middle contact (MC) is negatively biased, while the top (TC) and bottom contacts (BC) are positively biased.

As shown in Figure 4.15, first, the highly *p* doped, $In_{0.49}Ga_{0.51}P$ (lattice matched to GaAs) contact layer will be grown on GaAs substrate. Next the $Al_{0.8}Ga_{0.2}As$ barrier, $In_{0.49}Ga_{0.51}P$ emitter, and $Al_{0.8}Ga_{0.2}As$ barrier are designed to have a 0.088 eV barrier height, which is similar to the split-off energy of the $In_{0.49}Ga_{0.51}P$ emitter layer. This relatively high Al fraction should not introduce growth difficulties since AlGaAs with an 80% Al fraction has already been reported[153-158]. A highly doped, GaAs middle contact layer is followed by $Al_{0.57}Ga_{0.43}As$/*p*-GaAs/$Al_{0.57}Ga_{0.43}As$ (barrier/emitter/barrier) layers that are grown on top of the last barrier to achieve a 0.248 eV barrier.

4.8 Conclusion

An uncooled infrared detector based on a GaAs/AlGaAs multiple heterostructure was reported. The response is primarily from heavy/light hole to split-off transitions, and the detector showed a peak D^* of 6.8×10^5 Jones at 2.5 μm 300 K. As a well-developed material system, GaAs is a feasible solution to future uncooled infrared detection; hence high quality growth and integration with other readout electronics is readily available. Materials other than GaAs/AlGaAs may help to extend the coverage to longer wavelengths. Possible materials such as phosphides (with a threshold of ~18 μm) and nitrides may be able to operate at 60 μm or beyond at elevated temperatures. A possible dual-band detector design was proposed to cover 3-5 μm and 8-14 μm atmospheric windows using a combined system with arsenides and phosphides. The response can be optimized by using a graded-heating-barrier that reduces trapping and increases the gain. Another optimization method would be the use of surface plasmon resonances in metallic nanoparticles deposited on the detector surface[159]. A properly optimized device working at room temperature may compete with currently available uncooled detectors[160,161]. The long wavelength response observed in the split-off detector is due to a thermal mechanism. This mechanism involves a new type of bolometric response, based on the increase in carriers in the split-off band for *p*-type material. These carriers are free to be transported through the detector, producing the photo-signal. This is a new approach to bolometer development and could lead to improved detectors. By using other materials, the operating temperature can be optimized for the desired response wavelength range. These detectors offer an interesting new approach to MIR and FIR detection at high temperature, and could lead to many new applications.

Chapter 5

GaSb homojunctions for Far-IR (THz) detection

5.1 Introduction

Much of the recent interest in terahertz (THz) frequency radiation between 10 to 0.3 THz (30 μm to 1 mm) stems from its ability to penetrate many organic materials without the ionizing damage associated with short wavelength radiation such as X-rays[162]. Another important feature is the fact that THz radiation is readily absorbed by water. This allows THz radiation to distinguish between materials with varying water content, for example, fat versus lean meat. These properties offer various applications related to process and quality control, as well as biomedical imaging[30]. Moreover, tests are currently under way to determine whether THz tomographic imaging can augment or replace standard mammography. Others have proposed THz imaging as a method of screening passengers for explosives at airports. For military and astrophysics uses, THz detectors have many applications such as early detection of long range missiles, concealed weapon identification and space based astronomy. Those studies explain the reasons for the surging demand for high performance THz detectors in recent years.

Charge transport properties[163] and THz absorption[164] of high quality GaSb films were reported elsewhere. Infrared (IR) detectors based on type II InAs/GaSb superlattices[165] have been demonstrated for the 3-5 μm[166] and 8-12 μm[167] transparency bands of the atmosphere. The longest zero response threshold wavelength reported was 20 μm[168].

5.2 GaSb homojunction detector mechanism

The active region of a Homojunction Interfacial Workfunction Internal Photoemission (HIWIP) detector consists of one or more periods, each consisting of a series of doped emitter and undoped barrier layers. In HIWIP detectors, the incident photons are absorbed in the emitter layers by the free carrier absorption mechanism, and under suitable bias conditions the photoexcited carriers with energy $h\nu$ larger than the band-edge offset between the doped emitter layer and the undoped barrier region are emitted across the junction and then collected. For a homojunction, the band offset is determined by the doping-induced bandgap narrowing of the emitter[105]. The internal photoemission of the carriers is characterized by the interfacial work function Δ which corresponds to the energy difference between the bottom of the barrier (for a p-HIWIP) and the Fermi level in the emitter. In general these detectors show broad band response due to free carrier absorption which involves initial and final energy states within the same continuum. The zero-response threshold frequency, f_0 is introduced only in the photoemission stage. Therefore, the threshold frequency of the detector is $f_0 = 0.242\Delta$ with f_0 in THz and Δ in meV, or the zero-response threshold wavelength of the detector is $\lambda_0 = 1240/\Delta$ where λ_0 is in μm and Δ in meV.

5.3 Experimental Design

The HIWIP structure was grown on a GaSb substrate at a temperature of 600 °C and a reactor pressure of 400 Torr by metal-organic vapor phase epitaxy[169] using a Thomas Swan vertical reactor. Conventional trimethylgallium and trimethylantimony were used as precursors. The structure of the detector, grown on a 0.7 mm thick, 5×10^{18} cm^{-3} p-doped, GaSb substrate, consisted of a 0.05 μm thick, 2×10^{18} cm^{-3} p-doped, bottom emitter, a 2 μm thick undoped barrier, a 0.05 μm thick, 2×10^{18} cm^{-3} p-doped, top emitter, and a 0.1 μm thick, 5×10^{18} cm^{-3} p-doped, top contact. Zinc was the dopant, with diethylzinc used as the dopant source. The detector was fabricated by etching different size mesas using wet etching techniques. Ti/Pt/Au ohmic contacts were evaporated for both the top and bottom contacts. A schematic view of the processed detector and its relevant band diagram is shown in Figure 5.1.

5.3.1 Dark current, responsivity and noise measurements

Dark and 300 K background current voltage (IV) characteristics of the sample were measured at various temperatures using a Keithley 2400 source meter with the sample mounted on the cold finger of a closed cycle refrigerator. The spectral response measurements were carried out using a Perkin-Elmer System2000 Fourier-transform infrared (FTIR) spectrometer with a Si composite bolometer as the reference. The sample was mounted in a liquid He dewar made by Infra-Red Laboratory Inc. The same setup was connected to a SR-785, two-channel, fast Fourier transform signal analyzer coupled to a SR-560, high impedance, low-noise, voltage preamplifier for noise current density measurements.

5.4 Results and discussion

The sample GSU-A3 contains square mesas with sides of 400, 600, 800, and 1000 μm. The measured dark current density variation for 8 different mesas of the sample at 10 K is shown in Figure 5.2. Current densities are almost the same for different size mesas of untreated samples, showing that it is free of major side-wall/surface leakage.

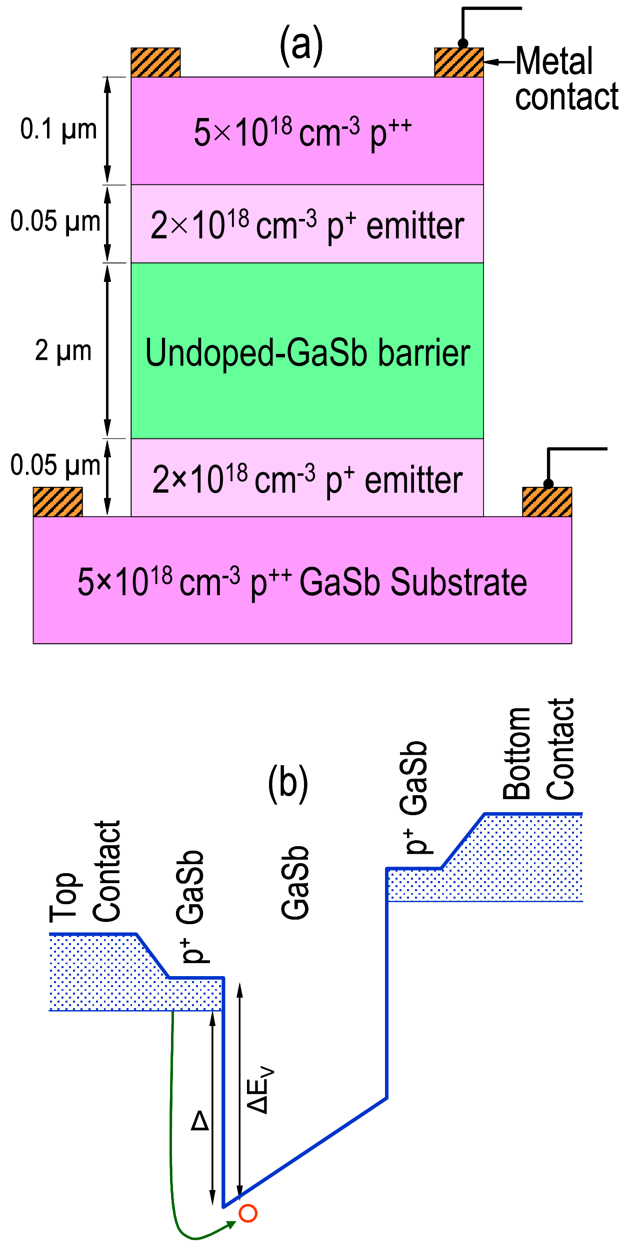


Figure 5.1 (a) Schematic drawing of the processed detector and (b) an energy-band diagram with the detection mechanism of the device under forward bias. The internal work function *Δ* originates from the doping offset between the emitter and the barrier. Here ΔE_V is the valence-band-edge offset at the p$^+$ and undoped interface.

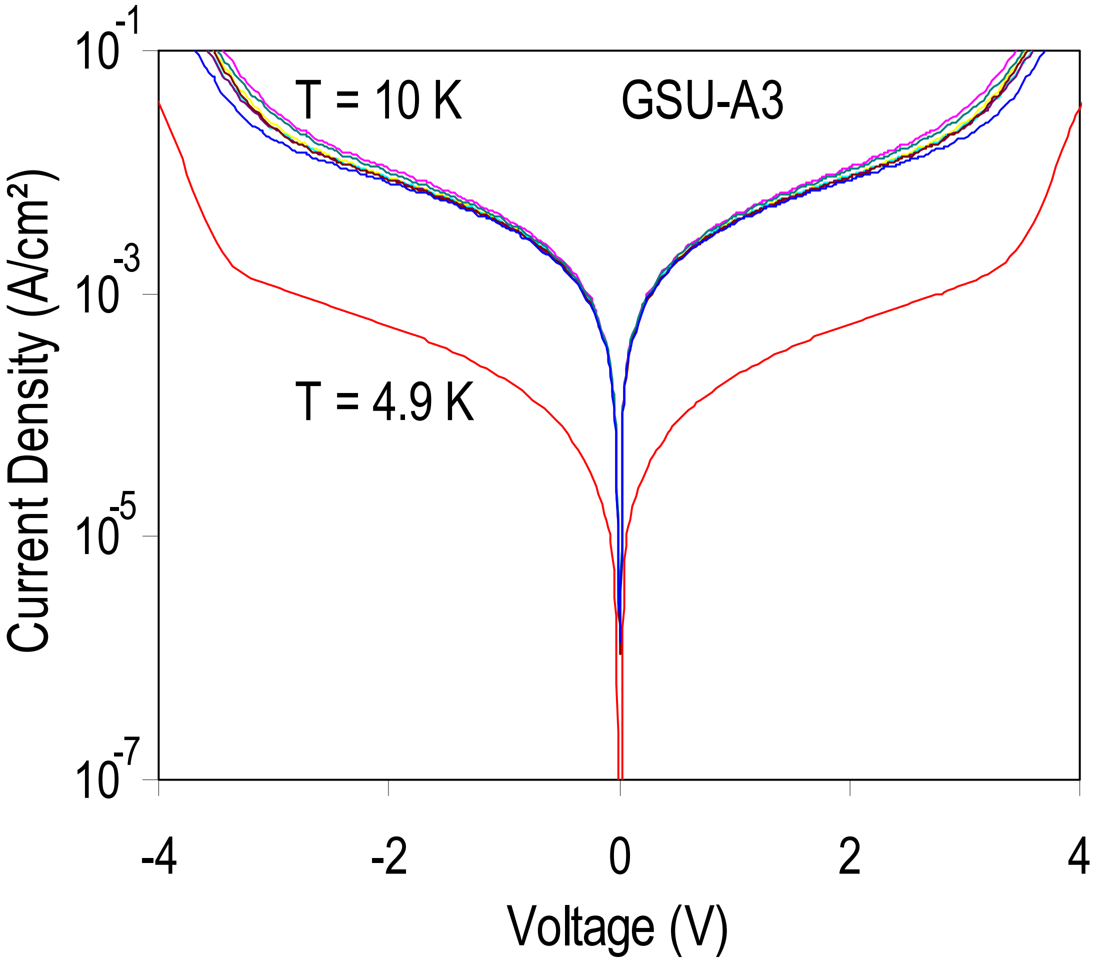


Figure 5.2 The dark current density vs. bias voltage for 8 different mesas in sample GSU-A3 measured at 10 K shows good uniformity of the structure. A single dark IV curve for one mesa at the 4.9 K spectral measurement temperature is also shown.

Normally larger devices (400 - 1000 μm) are not affected very much by surface leakage, which often occurs for devices less than about 50 × 50 μm. The dark current density for an 800 μm mesa at 4.9 K is also shown in Figure 5.2. The measured responsivity of GSU-A3 under different bias voltages and over the wavelength range 20-44 μm is shown in Figure 5.3. An increase in responsivity can be observed for applied bias increasing up to a field of 18.5 kV/cm (3.7 V), but above this field the responsivity started to saturate. The maximum response, 9.7 A/W, was obtained at 36 μm under a 3.7 V bias. The detector shows a maximum detectivity of 5.7×10^{11} Jones at a 3V bias. Above this voltage, the noise current is higher due to the low dynamic resistance of the sample, reducing the detectivity at a 3.7 V bias. The responsivity decreases with increasing operating temperature, and no significant signal could be measured above 15 K. The responsivity drops to zero around 43 μm due to the reststrahlen band of GaSb. The detector responsivity from 20 μm to 200 μm for 1V, 2V and 3V biases is shown in Figure 5.4. The corresponding background noise level (in the bottom of Figure 5.4) was measured after blocking the IR beam of the FTIR spectrometer by using a thick metal plate. Above the reststrahlen absorption minimum at 43 μm, the response initially increases and then gradually decays up to the 97 μm threshold. At lower frequencies, (longer wavelengths) the sample shows a clear but low flat response up to 1.5 THz (200 μm).

This response is relatively low compared with the peak response at 8.3 THz (36 μm) but clearly an order of magnitude above the noise level. The 20 μm to 43 μm photoresponse does not exhibit any dependence on measurement speed (optical path difference velocity of FTIR) however the constant long wavelength tail (43 – 200 μm) response increases with decreasing measurement speed. A second sample processed in a different laboratory displayed a similar flat response long wavelength tail, confirming that it is not processing or contact related.

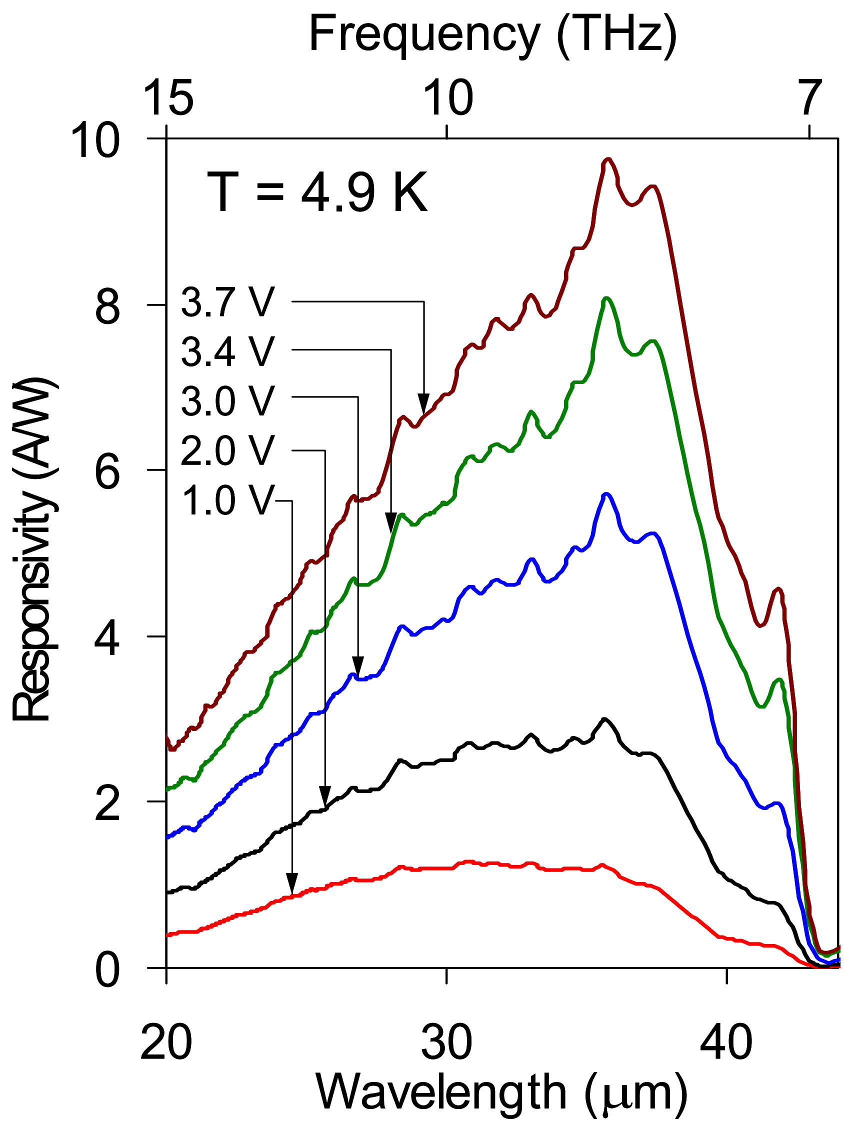


Figure 5.3 Spectral response as measured at 4.9 K for sample GSU-A3 under different forward bias voltages. The 20 to 43 μm range responsivity has a peak responsivity of 9.7 A/W at 36 μm and 3.7 V bias.

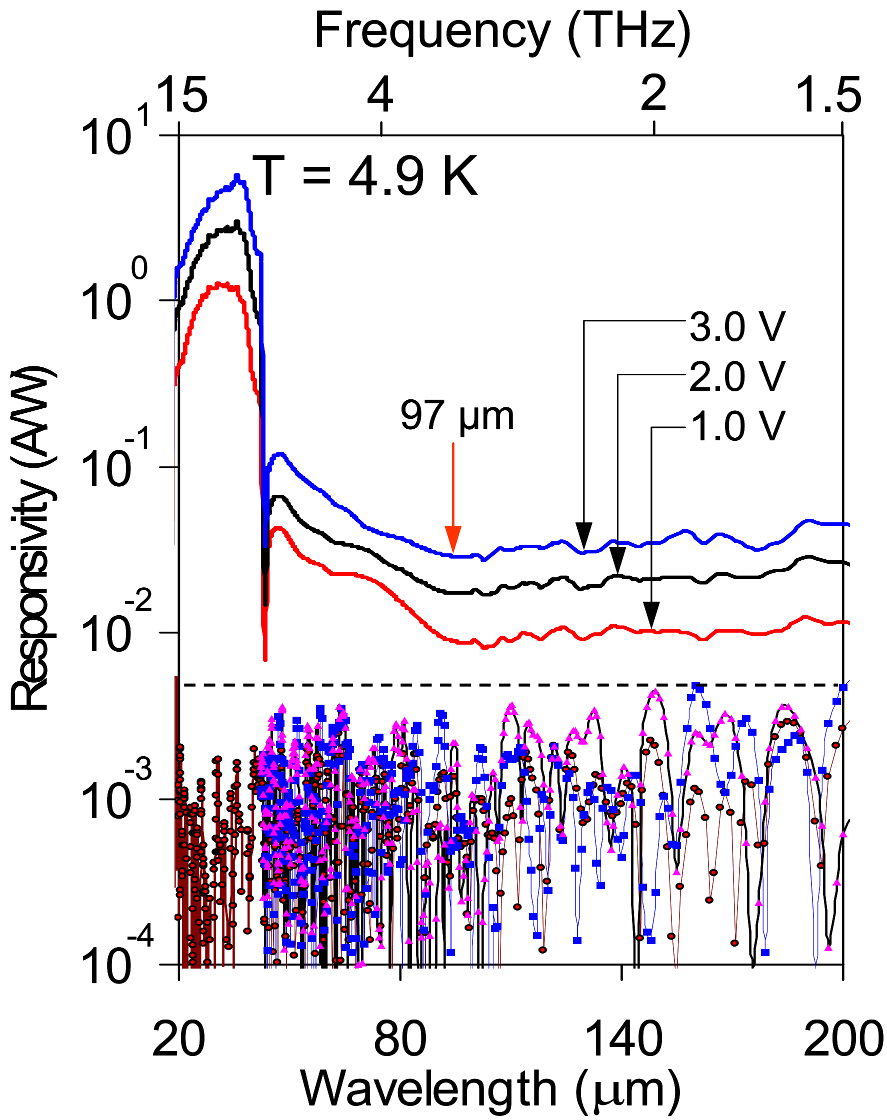


Figure 5.4 The full spectrum including the long wavelength response at 1, 2 and 3 V bias conditions. Note that the response is plotted in a log scale in order to emphasize the long wavelength response. The arrow shows the 97 μm free carrier response threshold wavelength position. The noise curves at the bottom were taken by blocking the FTIR source beam and the dashed line shows the maximum noise level of the spectral response measurement setup.

The zero response threshold wavelength λ_0 of the detector is not clearly visible, since it corresponds to two different overlapping response mechanisms. The free carrier threshold of λ_0 = 97 μm (arrow in Figure 5.4) was obtained by selecting the intersection point of the flat long wavelength response with the free carrier response. This value agrees well with the calculated Arrhenius Δ value. Figure 5.5 shows the $\ln(I/T^{1.5})$ vs. $1/T$ plot that was used to calculate the internal workfunction Δ. The long wavelength flat response mechanism is not yet clearly understood.

The response time of the detector can be estimated from the bias dependent responsivity measurements[170,171]. For low experimental temperatures (T ~ 4.2 K), the current responsivity R at a bias voltage V can be expressed as

$$R = \frac{3q\eta\mu\tau}{2d^2}\frac{1}{k_B T_0}V \tag{5.1}$$

where q is elementary charge, η is the absorption quantum efficiency, μ the mobility, k_B is the Boltzmann constant, d the detector thickness, T_0 is the equilibrium temperature and τ is the energy relaxation time of the hot carriers. The energy relaxation time could give the lower limit for the detector response time, and could be regarded as the response time for an ideal detector. The bias dependence of the responsivity measured at 4.9 K is shown in Figure 5.6. At low voltages (V < 2.5V) the responsivity increases linearly with bias. For both intrinsic and extrinsic photoconductive detectors at a given bias, the responsivity is proportional to the response time. The absorption quantum efficiency η can be expressed as described in Ref [170] as

$$\eta = \frac{1-e^{2\alpha l}}{e^{(d/L_Z)}} \; . \tag{5.2}$$

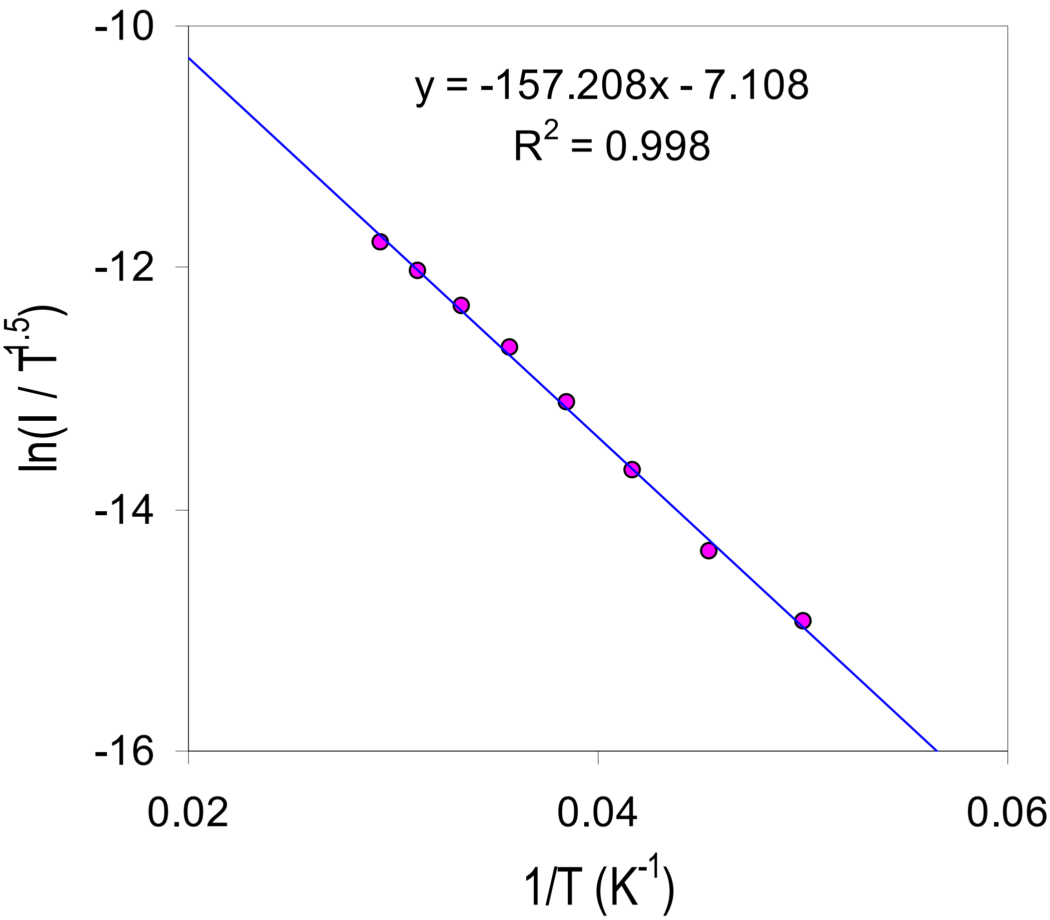


Figure 5.5 Arrhenius curve of the sample showing slope of -157.2. The internal workfunction Δ of 12.8 ± 0.3 meV was calculated using this slope.

Here α, l and L_z are the absorption coefficient, thickness of the emitter layer, and inelastic scattering mean free path, respectively. Using the GaSb α from Ref. [164] and taking L_z as 1 μm, the total absorption quantum efficiency was calculated as 17.3%. Using a detector thickness d of 2.1 μm, the measured mobility[164] of 200 cm^2/Vs for similar doping levels, and the slope of 1.6 A/WV measured from the linear region (0 - 2.5 V) of R vs. V plot, the response time of the detector is estimated to be of the order of a few picoseconds. The photocurrent gain values g are calculated for the peak response (36 μm) as 0.6 at 2 V and 1.9 at 3.7 V biases from the combination of $R=q\eta g\lambda/hc$ with Eq.(5.2) , where h, c, and λ are the Planck constant, speed of light, and wavelength at peak response. The effective quantum efficiency ηg was calculated as 10 and 33 for 2 V and 3.7 V biases, respectively. The above photocurrent gain numbers and calculated shot noise gain[104] numbers differ for this sample. This difference and the unusually high impedance of the sample are believed to be due to the Schottky type contacts[172] . Further studies are needed to examine the effects of Schottky contacts on HIWIP detectors.

5.5 Conclusion

GaSb based, HIWIP detectors with a 97 μm (~3 THz) zero response threshold wavelength were demonstrated. The detectors have a peak responsivity of 9.7 A/W and peak detectivity of 5.7×10^{11} Jones at 4.9 K. In this study a simple single emitter / barrier structure was utilized. The high response observed could be further enhanced by using multilayer structures which will increase the photon absorption efficiency and could possibly induce photocurrent gain enhancement[173]. The doping offset of the emitter and barrier determined the threshold of the HIWIP detector. Using Heterojunction Interfacial Workfunction Internal Photoemission (HEIWIP) structures[104] like $In_xGa_{1-x}Sb$/GaSb would give a flexible system to control the threshold.

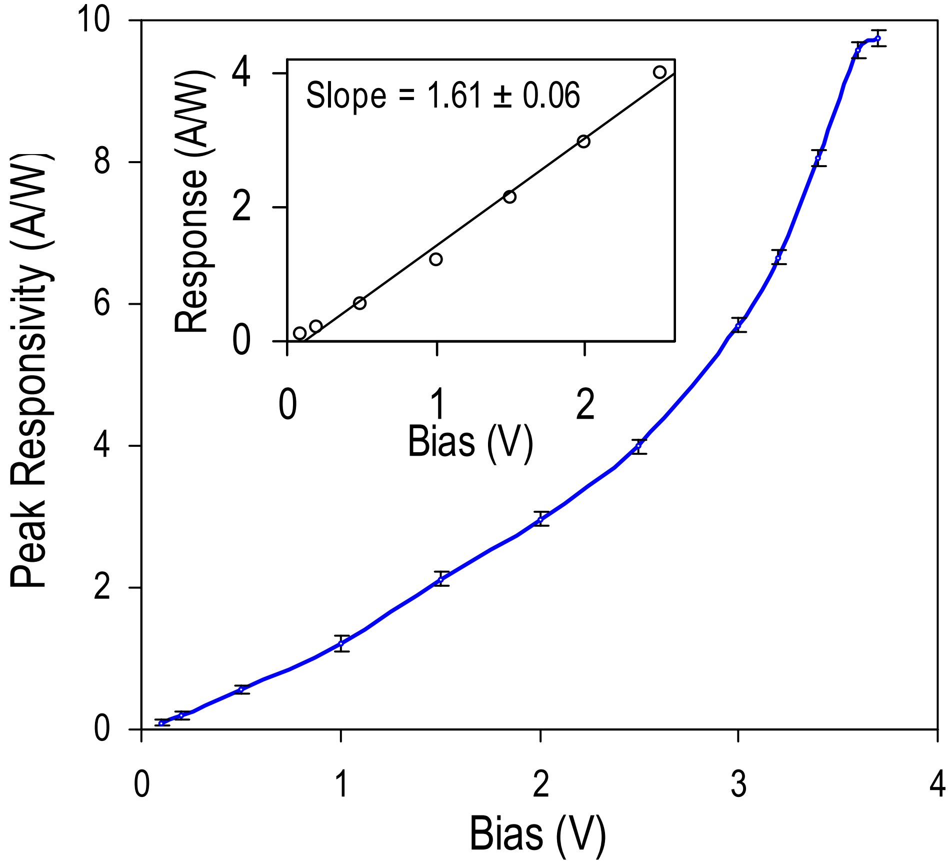


Figure 5.6 The bias dependence of the peak responsivity at 4.9 K. The responsivity was almost linear (with a slope of 1.61 ± 0.06 A/WV) for low biases (< 2.5 V).

Estimations show that with the variation of the In fraction, a 1 THz (300 μm) threshold could be achieved while keeping the accuracy of the In fraction in the practical range (a variation of 2.5% will only change the workfunction by 1 meV). One of the major problems in achieving a 1 THz threshold with GaAs/AlGaAs or GaN/AlGaN system is the need for very high Al fraction accuracy (0.2% for AlGaAs and 0.05% for AlGaN) that is at or beyond the edge of the experimental capabilities. Therefore, GaSb/InGaSb is a promising material for THz detection. Resonant cavity enhanced GaAs/AlGaAs HEIWIP's have already been demonstrated[174] and resonant cavities can be easily incorporated into GaSb based detectors to further enhance the response.

Chapter 6

Summary and conclusion

Here relative merits of each of the techniques discussed before are presented. The four chapters in the dissertation presented four different infrared detection concepts with the experimental results to verify each concept. First two chapters discussed an amorphous semiconductor (TiO_2, CuSCN, and PbS) based devices and the other two chapters discussed detection based on epitaxially grown crystalline semiconductors (GaAs, AlGaAs, and GaSb). All experimental results presented in previous chapters are summarized in Table 6.1.

The detectors based on dye-sensitized nanostructures and colloidal quantum dots utilized simple growing techniques such as screen printing or spray pyrolysis hence relatively inexpensive and can also be fabricated on curved surfaces or flexible substrates. DSNIRD's followed the same principle as DSSC known as Gratzel's solar cell but in this case, the liquid electrolyte was replaced with a solid hole collecting (p-type) semiconductor CuSCN. Commercially available IR sensitive dyes used in these initial experiments are not optimized for dye-sensitization. Not only the HOMO and LOMO levels of the dye but also the bonding and charge transfer properties of the dye with the semiconductor layers play major roll on the conversion efficiency of the device.

Table 6.1 A summary of selected detectors demonstrated in this work along with their specifications. λ_p, λ_{Range}, R_p, D*, and T stand for peak wavelength, response wavelength range, peak responsivity, specific detectivity, operating temperature respectively.

Structure Type	Sample	λ_{Range}(peak) (μm)	R_p (mA/W)	Conversion Efficiency (%)	D* @ T (Jones)	Operating Temperature (K)
DSNIRD[21]	IR783	0.4 - 0.9 (0.808)	0.4	0.06	1.4×10^{10}	300
	MC-IR792	0.4 - 0.95(0.812)	2.7	0.41	9.5×10^{10}	300
	IR820-IR1040	0.4 - 1.1 (0.858)	1.0	0.14	3.5×10^{10}	300
	IR820	0.4 - 0.95(0.866)	0.3	0.04	1.1×10^{10}	300
	BPR-IR820	0.4 - 0.97(0.876)	1.1	0.16	3.9×10^{10}	300
	BPR-IR1040	0.4 - 1.2 (1.056)	0.3	0.04	1.1×10^{10}	300
CQD[25]	PbS-300	0.4 -1.1(540)	195 V/W	-	3.0×10^{8}	300
Split-off[24]	SP1	1.5 - 4 (2.5)	2.3	0.11	2.1×10^{6}	140
	SP2	1.5 - 4 (2.5)	2.7	0.13	1.8×10^{6}	190
	SP3	1.5 - 4 (2.5)	0.29	0.01	6.8×10^{5}	300
GaSb HIWIP[26]	GSU-A3	20 - 97(36)	9700	33	5.7×10^{11}	4.9

This study was not intended to modify the chemical structure and synthesis optimized dyes for IR range sensitization. Instead, several different anionic compounds were used to improve the binding property of cationic dyes to the semiconductor. The observed peak conversion efficiency ~0.4% at 812 nm is relatively low compared to properly optimized dyes for visible wavelength such as Ru-N3 with experimentally proven 10% conversion efficiency. DSCS is a rapidly developing research area and new findings in DSSC field can be directly incorporate to improve the performance of DSNIRD's. These detectors have the advantage of not being sensitive to thermal noise and radiation that initiate bandgap excitations. Despite quite low conversion efficiency the peak specific detectivity (9.5×10^{10} Jones at 812 nm) measured was comparable to commercially available Si or InGaAs photodiodes due to the low noise level of the DSNIRD's. Due to the dye degradation especially when UV radiation is present, the life time of the DSNIRD's is low compared to Si or InGaAs. Properly sealed detectors eliminate moister with a UV protective coating on front window will resolve the two issues.

PbS CQD embedded capacitor type detector demonstrate during this study reveal a novel detection mechanism without direct contact to QD or quantum wires which shows optically generated electronic polarizability. Low cost fabrication process and wavelength tunability by changing size of the QD, and fabrication possibility on flexible substrates are the attractive features of these detectors. However observed detectivity values are two orders of magnitude lower than currently available detectors. In order to increase responsivity QD density inside the capacitor should be increased. However in general conductor-insulator composites exhibit percolation thresholds when the packing fraction of the conducting material exceeds a critical value. Therefore QD density cannot be increased arbitrally. CQD capacitor based detector needed an optical chopper or modulated incident radiation since these detectors are sensitive only to a change in the intensity.

This is not practical for some applications. Finally the growing techniques of DSNIRD's and CQD based detectors are not directly compatible with standard Si readout electronic processing, direct integration may not be possible at the current state-of-the-art. Further research will be needed to develop detector arrays and integration methods. In other words both detectors are more suitable for low cost large area single element detectors instead of detector arrays for imaging applications.

GaAS/AlGaAs heterojunction based split-off band detectors and GaSb homojunction based THz detector discussed in chapter 4 and 5 are III-V semiconductor based structures grown by molecular beam epitaxy, or metalorganic chemical vapor deposition techniques. Split-off band detector utilize the heavy hole light hole transition for response and the free carrier threshold for perform tailoring. The split-off transition will be fixed for a particular material. Different materials can give rise to different response regions, due to the split-off energy differences. Using bandgap engineered III-V materials such as InGaP and InGaAsP, split-off detectors could be develop covering the wavelength ranges from 3-12 microns. This approach gives the additional controllability of the operating temperature, responsivity and specific detectivity by tuning the free carrier threshold. Three different set of samples with λ_t to split-off energy ratio 42%, 57%, and 85% (sample SP1, SP2, SP3 respectively) were tested during this study. Operating temperature increased with λ_t to split-off energy ratio as expected. The highest operating temperature of 330 K was observed for SP3 with an 85% λ_t to split-off energy ratio. According to those results 77% λ_t to split-off energy ratio will be the best for 300 K room temperature operation. Room temperature detectivity observed in the sample SP3 is quite low compared to available competitors such as InGaAs photodiodes. As a well-developed material system, GaAs is a feasible solution to future uncooled infrared detection; hence high quality growth and integration with other readout electronics is readily available.

GaSb has a higher free carrier absorption coefficient at THz frequencies compared to the GaAs. Therefore it is a promising material for THz detectors. GaSb /InGaSb HEIWIP detector have advantage over GaAs/AlGaAs system of tailoring λ_t to very long wavelengths still keeping a higher In fraction which is not possible with GaAs/AlGaAs system. GaSb is not a well developed material system such as GaAs, Si, or Ge. A GaSb homojunction based THz detector was analyzed as an initial study to test the quality of the material and processing. The first set of samples showed (see the Table 6.1) comparable detection parameters with available other detectors such as HgCdTe and Si composite bolometers. Both the split-off band detectors and GaSb HIWIP detectors shows long wavelength flat response which is not expected at from the free carrier response. This can be identified as a bolometric type response since the response is highly measurement speed dependent. Future research needed to optimized these response and compare with available bolometers.

BIBLIOGRAPHY

1 C. A. Curcio, K. R. Sloan, R. E. Kalina, and A. E. Hendrickson, "Human photoreceptor topography", The Journal of Comparative Neurology **292,** 497-523 (1990).

2 R. N. Clark, "Notes on the Resolution and Other Details of the Human Eye", http://www.clarkvision.com/imagedetail/eye-resolution.html, (2005)

3 K. V. Kardong and S. P. Mackessy, "The Strike Behavior of a Congenitally Blind Rattlesnake", Journal of Herpetology **25,** 208-211 (1991).

4 A. R. Krochmal, G. S. Bakken, and T. J. LaDuc, "Heat in evolution's kitchen: evolutionary perspectives on the functions and origin of the facial pit of pitvipers (Viperidae: Crotalinae)", Journal of Experimental Biology **207,** 4231-4238 (2004).

5 W. Herschel, "Investigation of the powers of the prismatic colours to heat and illuminate objects; with remarks, that prove the different refrangibility of radiant heat. To which is added, an inquiry into the method of viewing the Sun advantageously, with telescopes of large apertures and high magnifying powers." Philosophical Transactions of the Royal Society of London, Series A: Physical Sciences and Engineering **90,** 255–283 (1800).

6 W. Herschel, "Experiments on the Refrangibility of the Invisible Rays of the Sun", Philos. Trans. Roy. Soc. Londin **90,** 284–292 (1800).

7 A. Rogalski, H. Mohamed, and R. Manijeh, in *Handbook of Infra-red Detection Technologies* (Elsevier Science, Amsterdam, 2002), p. 5-81.

8 C. V. Hoof, P. D. Moor, H. Mohamed, and R. Manijeh, in *Handbook of Infra-red Detection Technologies* (Elsevier Science, Amsterdam, 2002), p. 449-479.

[9] G. Dall'Oglio, B. Melchiorri, F. Melchiorri, and V. Natale, "Comparison between carbon, silicon and germanium bolometers and Golay cell in the far infrared", Infrared Physics **14,** 347-350 (1974).

[10] K. Yamashita, A. Murata, and M. Okuyama, "Miniaturized infrared sensor using silicon diaphragm based on Golay cell", Sensors and Actuators A: Physical **66,** 29-32 (1998).

[11] P. W. Steinhage and R. R. Zeyfang, "Performance of pyroelectric detectors and vidicons based on modified Pb(Ti, Zr)O_3 ceramics", Ferroelectrics **11,** 301 - 304 (1976).

[12] A. Rogalski, "Infrared detectors: status and trends", Progress in Quantum Electronics **27,** 59-210 (2003).

[13] A. Piotrowski, P. Madejczyk, W. Gawron, K. Klos, J. Pawluczyk, J. Rutkowski, J. Piotrowski, and A. Rogalski, "Progress in MOCVD growth of HgCdTe heterostructures for uncooled infrared photodetectors", Infrared Physics & Technology **49,** 173-182 (2007).

[14] A. G. U. Perera, W. Z. Shen, S. G. Matsik, H. C. Liu, M. Buchanan, and W. J. Schaff, "GaAs/AlGaAs quantum well photodetectors with a cutoff wavelength at 28 μm", Appl. Phys. Lett. **72,** 1596-1598 (1998).

[15] A. Rogalski, "Comparison of the performance of quantum well and conventional bulk infrared photodetectors", Infrared Physics & Technology **38,** 295-310 (1997).

[16] D. D. Coon and R. P. G. Karunasiri, "New Mode of IR Detection Using Quantum Wells", Appl. Phys. Lett. **45,** 649-651 (1984).

[17] A. G. U. Perera, H. X. Yuan, and M. H. Francombe, "Homojunction internal photoemission far-infrared detectors: Photoresponse performance analysis", J. Appl. Phys. **77,** 915-924 (1995).

[18] A. G. U. Perera, S. G. Matsik, B. Yaldiz, H. C. Liu, A. Shen, M. Gao, Z. R. Wasilewski, and M. Buchanan, "Heterojunction wavelength-tailorable far-infrared photodetectors with response out to 70 μm", Appl. Phys. Lett. **78,** 2241-2243 (2001).

[19] A. L. Korotkov, A. G. U. Perera, W. Z. Shen, J. Herfort, K. H. Ploog, W. J. Schaff, and H. C. Liu, "Free-carrier absorption in Be- and C-doped GaAs epilayers and far infrared detector applications", Journal of Applied Physics **89,** 3295-3300 (2001).

[20] A. G. U. Perera, in *International Bureau of WIPO*; *Vol. WO 2008/028181* (U. S. Provisional Patent, Pending).

[21] P. V. V. Jayaweera, A. G. U. Perera, M. K. I. Senevirathna, P. K. D. D. P. Pitigala, and K. Tennakone, "Dye-sensitized near-infrared room-temperature photovoltaic photon detectors", Appl. phys. Lett. **85,** 5754-5756 (2004).

[22] A. G. U. Perera, S. G. Matsik, P. V. V. Jayaweera, K. Tennakone, H. C. Liu, M. Buchanan, G. Von Winckel, A. Stintz, and S. Krishna, "High operating temperature split-off band infrared detectors", Applied Physics Letters **89,** 131118-3 (2006).

[23] P. V. V. Jayaweera, S. G. Matsik, K. Tennakone, A. G. U. Perera, H. C. Liu, and S. Krishna, "Spin split-off transition based IR detectors operating at high temperatures", Infrared Physics & Technology **50,** 279-283 (2007).

[24] P. V. V. Jayaweera, S. G. Matsik, A. G. U. Perera, H. C. Liu, M. Buchanan, and Z. R. Wasilewski, "Uncooled infrared detectors for 3-5 μm and beyond", Applied Physics Letters **93,** 021105-3 (2008).

[25] P. V. V. Jayaweera, A. G. U. Perera, and K. Tennakone, "Displacement currents in semiconductor quantum dots embedded dielectric media: A method for room temperature photon detection", Appl. Phys. Lett. **91,** 063114-3 (2007).

[26] P. V. V. Jayaweera, S. G. Matsik, A. G. U. Perera, Y. Paltiel, A. Sher, A. Raizman, H. Luo, and H. C. Liu, "GaSb homojunctions for far-infrared (terahertz) detection", Appl. Phys. Lett. **90,** 111109-3 (2007).

[27] P. V. V. Jayaweera, P. K. D. D. P. Pitigala, M. K. I. Seneviratne, A. G. U. Perera, and K. Tennakone, "1/f Noise in dye-sensitized solar cells and NIR photon detectors", Infrared Physics & Technology **50,** 270-273 (2007).

[28] P. V. V. Jayaweera, P. K. D. D. P. Pitigala, A. G. U. Perera, and K. Tennakone, "1/f noise and dye-sensitized solar cells", Semicond. Sci. Technol. **20,** L40–L42 (2005).

[29] D. Adrian, O. Chiko, and K. Kodo, "Terahertz-wave sources and imaging applications", Measurement Science and Technology**,** R161 (2006).

[30] P. H. Siegel, "Terahertz technology in biology and medicine", Microwave Theory and Techniques, IEEE Transactions on **52,** 2438-2447 (2004).

[31] S. A. Harmon and R. A. Cheville, "Part-per-million gas detection from long-baseline THz spectroscopy", Applied Physics Letters **85,** 2128-2130 (2004).

[32] W. H. F. Talbot, "An Account of the Processes employed in Photogenic Drawing", Proceedings of the Royal Society of London, Series A: Mathematical and Physical Sciences **4,** 124-126 (1839).

[33] B. O'Regan and M. Gratzel, "A low-cost, high-efficiency solar cell based on dye-sensitized colloidal TiO_2 films", Nature **353,** 737-740 (1991).

[34] H. C. Liu, J. -P., Noel, Lujian, Li, M. Buchanan, and J. G. Simmons, "Novel infrared detector concept utilizing controlled epitaxial doping profiles", Appl. Phys. Lett. **60,** 3298-3300 (1992).

[35] K. K. Choi, G. Dang, J. W. Little, K. M. Leung, and T. Tamir, "Quantum grid infrared spectrometer", Appl. Phys. Lett. **84,** 4439-4441 (2004).

[36] K. Stewart, M. Buda, J. Wong-Leung, L. Fu, C. Jagadish, A. Stiff-Roberts, and P. Bhattacharya, "Influence of rapid thermal annealing on a 30 stack InAs/GaAs quantum dot infrared photodetector", Journal of Applied Physics **94,** 5283-5289 (2003).

[37] S. Krishna, S. Raghavan, G. von Winckel, P. Rotella, A. Stintz, C. P. Morath, D. Le, and S. W. Kennerly, "Two color InAs/InGaAs dots-in-a-well detector with background-limited performance at 91 K", Appl. Phys. Lett. **82,** 2574-2576 (2003).

[38] A. Roglaski, "Infrared Detectors: anoverview", IR Physics and Tech. **43,** 187-210 (2002).

[39] G. Smestad, *Optoelectronics of Solar Cells* (SPIE, Bellingham, WA, 2002).

[40] K. Tennakone, G. R. R. A. Kumara, A. R. Kumarasinghe, K. G. U. Wijayantha, and P. M. Sirimanne, "A dye-sensitized nano-porous solid-state photovoltaic cell", Semiconductor Science and Technology **10,** 1689-1693 (1995).

[41] B. O'Regan and D. T. Schwartz, "Efficient Photo-Hole Injection from Adsorbed Cyanine Dyes into Electrodeposited Copper(I) Thiocyanate Thin Films", Chemistry of Materials **7,** 1349-1354 (1995).

[42] G. R. R. A. Kumara, A. Konno, G. K. R. Senadeera, P. V. V. Jayaweera, D. B. R. A. De Silva, and K. Tennakone, "Dye-sensitized solar cell with the hole collector p-CuSCN deposited from a solution in n-propyl sulphide", Solar Energy Materials and Solar Cells **69,** 195-199 (2001).

[43] P. K. D. D. P. Pitigala, M. K. I. Seneviratne, V. P. S. Perera, and K. Tennakone, "Sensitization of Nanostructured TiO_2 by Electrostatic Coupling of Ionic Dyes to Ionic Absorbates", Langmuir **20,** 5100-5103 (2004).

[44] V. P. S. Perera, P. K. D. D. P. Pitigala, P. V. V. Jayaweera, K. M. P. Bandaranayake, and K. Tennakone, "Dye-Sensitized Solid-State Photovoltaic Cells Based on Dye Multilayer-Semiconductor Nanostructures", Journal of Physical Chemistry B **107,** 13758-13761 (2003).

[45] M. K. Nazeeruddin and M. Gratzel, *Semiconductor Electrodes and Photoelectrochemistry* (Wiley, VCH, New York, 2002).

[46] N. Sclar, "Properties of doped silicon and Germanium infrared detectors", Progress in Quantum Electronics **9,** 149-257 (1984).

[47] A. Balandin, K. Wang, S. Cai, R. Li, C. Viswanathan, E. Wang, and M. Wojtowicz, "Investigation of flicker noise and deep-levels in GaN/AlGaN transistors", Journal of Electronic Materials **29,** 297-301 (2000).

[48] A. Balandin, *Noise and Fluctuation Control in Electronic Devices* (American Scientific, Los Angeles, 2002).

[49] S. Uchida, M. Tomiha, H. Takizawa, and M. Kawaraya, "Flexible dye-sensitized solar cells by 28 GHz microwave irradiation", Journal of Photochemistry and Photobiology A: Chemistry **164,** 93-96 (2004).

[50] P. M. Jayaweera, R. M. S. P. Rajapaksha, and K. Tennakone, "TiO2 nano-porous photoelectrochemical cells (PECs) sensitized with mixed cationic/anionic dye systems: Role of the second cationic fluorescent dye on the photocurrent enhancement", Chemical Physics Letters **412,** 29-34 (2005).

[51] K. Tennakone, G. K. R. Senadeera, and P. V. V. Jayaweera, "Enhancement of the energy and quantum conversion efficiencies of a photoelectrochemical cell sensitized with a combination of cationic and anionic dyes", Current Science **81,** 76-77 (2001).

[52] M. Gratzel, "Conversion of sunlight to electric power by nanocrystalline dye-sensitized solar cells", Journal of Photochemistry and Photobiology A: Chemistry **164,** 3-14 (2004).

[53] K. Tennakone, G. R. R. A. Kumara, I. R. M. Kottegoda, K. G. U. Wijayantha, and V. P. S. Perera, "A solid-state photovoltaic cell sensitized with a ruthenium bipyridyl complex", Journal of Physics D: Applied Physics**,** 1492-1496 (1998).

[54] M. B. Weissman, "1f noise and other slow, nonexponential kinetics in condensed matter", Reviews of Modern Physics **60,** 537 (1988).

[55] A. K. Raychaudhuri, "Measurement of 1/f noise and its application in materials science", Current Opinion in Solid State and Materials Science **6,** 67-85 (2002).

[56] M. Ashby, N. Gordon, C. Elliott, C. Jones, C. Maxey, L. Hipwood, and R. Catchpole, "Investigations into the source of 1/f noise in HgxCd1−xTe diodes", Journal of Electronic Materials **33,** 757-760 (2004).

[57] H. Wong, "Low-frequency noise study in electron devices: review and update", Microelectronics Reliability **43,** 585-599 (2003).

[58] A. A. Balandin, *Noise and Fluctuations Control in Electronic Devices* (American Scientific Publishers, Los Angeles, 2002).

[59] F. N. Hooge, "1/ f noise is no surface effect", Physics Letters A **29,** 139-140 (1969).

[60] J. Pelz and J. Clarke, "Quantitative "local-interference" model for 1f noise in metal films", Physical Review B **36,** 4479 (1987).

[61] T. Hwa and M. Kardar, "Dissipative transport in open systems: An investigation of self-organized criticality", Physical Review Letters **62,** 1813 (1989).

[62] T. Dilmi, A. Chovet, and P. Viktorovitch, "Influence of a magnetic field on 1/f noise in ambipolar semiconductors: Evidence of its surface origin", Journal of Applied Physics **50,** 5348-5351 (1979).

[63] J. Nelson, "Continuous-time random-walk model of electron transport in nanocrystalline TiO2 electrodes", Physical Review B **59,** 15374 (1999).

[64] P. E. de Jongh and D. Vanmaekelbergh, "Investigation of the Electronic Transport Properties of Nanocrystalline Particulate TiO2 Electrodes by Intensity-Modulated Photocurrent Spectroscopy", J. Phys. Chem. B **101,** 2716-2722 (1997).

[65] H. Wang, J. He, G. Boschloo, H. Lindstrom, A. Hagfeldt, and S. E. Lindquist, "Electrochemical Investigation of Traps in a Nanostructured TiO2 Film", J. Phys. Chem. B **105,** 2529-2533 (2001).

[66] A. M. Eppler, I. M. Ballard, and J. Nelson, "Charge transport in porous nanocrystalline titanium dioxide", Physica E: Low-dimensional Systems and Nanostructures **14,** 197-202 (2002).

[67] G. K. Boschloo, A. Goossens, and J. Schoonman, "Photoelectrochemical Study of Thin Anatase TiO_2 Films Prepared by Metallorganic Chemical Vapor Deposition", Journal of the Electrochemical Society **144,** 1311-1317 (1997).

[68] P. J. D. Lindan, N. M. Harrison, and M. J. Gillan, "Mixed Dissociative and Molecular Adsorption of Water on the Rutile (110) Surface", Physical Review Letters **80,** 762 (1998).

[69] S. Yanagida,15th International Conference on Photochemical Conversion and Storage of Solar Energy, Paris, 2004.

[70] M. K. Nazeeruddin, S. M. Zakeeruddin, J. J. Lagref, P. Liska, P. Comte, C. Barolo, G. Viscardi, K. Schenk, and M. Graetzel, "Stepwise assembly of amphiphilic ruthenium

sensitizers and their applications in dye-sensitized solar cell", Coordination Chemistry Reviews **248,** 1317-1328 (2004).

71 J. Moser, S. Punchihewa, P. P. Infelta, and M. Graetzel, "Surface complexation of colloidal semiconductors strongly enhances interfacial electron-transfer rates", Langmuir **7,** 3012-3018 (1991).

72 A. D. Yoffe, "Low-dimensional systems: quantum size effects and electronic properties of semiconductor microcrystallites (zero-dimensional systems) and some quasi-two-dimensional systems", Advances in Physics **51,** 799 - 890 (2002).

73 T. J. Bukowski and J. H. Simmons, "Quantum Dot Research: Current State and Future Prospects", Critical Reviews in Solid State and Material Sciences **27,** 119 (2002).

74 R. Agarwal and C. M. Lieber, "Semiconductor nanowires: optics and optoelectronics", Applied Physics A: Materials Science & Processing **85,** 209-215 (2006).

75 H. C. Liu, M. Gao, J. McCaffrey, Z. R. Wasilewski, and S. Fafard, "Quantum dot infrared photodetectors", Appl. Phys. Lett **78,** 79 (2001).

76 K. L. Wang, S. Tong, and H. J. Kim, "Properties and applications of SiGe nanodots", Materials Science in Semiconductor Processing **8,** 389-399 (2005).

77 D. Pal, J. Walker, and E. Towe, "InAs/GaAs quantum-dot infrared photodetectors grown by molecular beam epitaxy", JVST B - Microelectronics and Nanometer Structures **24,** 1532-1535 (2006).

78 S. Chakrabarti, A. D. Stiff-Roberts, P. Bhattacharya, S. Gunapala, S. Bandara, S. B. Rafol, and S. W. Kennerly, "High-temperature operation of InAs-GaAs quantum-dot infrared photodetectors with large responsivity and detectivity", Photonics Technology Letters, IEEE **16,** 1361-1363 (2004).

[79] H. Lim, S. Tsao, W. Zhang, and M. Razeghi, "High-performance InAs quantum-dot infrared photodetectors grown on InP substrate operating at room temperature", Appl. Phys. Lett. **90,** 131112-3 (2007).

[80] S. Krishna, S. Raghavan, G. von Winckel, A. Stintz, G. Ariyawansa, S. G. Matsik, and A. G. U. Perera, "Three-color ($\lambda_{p1} \sim 3.8$, $\lambda_{p2} \sim 8.5$, and $\lambda_{p3} \sim 23.2$ μm) InAs/InGaAs quantum-dots-in-a-well detector", Appl. Phys. Lett. **83,** 2745-2747 (2003).

[81] G. Konstantatos, I. Howard, A. Fischer, S. Hoogland, J. Clifford, E. Klem, L. Levina, and E. H. Sargent, "Ultrasensitive solution-cast quantum dot photodetectors", Nature **442,** 180-183 (2006).

[82] S. A. McDonald, P. W. Cyr, L. Levina, and E. H. Sargent, "Photoconductivity from PbS-nanocrystal/semiconducting polymer composites for solution-processible, quantum-size tunable infrared photodetectors", Applied Physics Letters **85,** 2089-2091 (2004).

[83] N. C. Greenham, X. Peng, and A. P. Alivisatos, "Charge separation and transport in conjugated-polymer/semiconductor-nanocrystal composites studied by photoluminescence quenching and photoconductivity", Physical Review B **54,** 17628 (1996).

[84] E. Arici, N. S. Sariciftci, and D. Meissner, "Hybrid Solar Cells Based on Nanoparticles of CuInS2 in Organic Matrices", Advanced Functional Materials **13,** 165-171 (2003).

[85] A. Hossain and M. H. Rashid, "Pyroelectric detectors and their applications", Industry Applications, IEEE Transactions on **27,** 824-829 (1991).

[86] M. Sabathil, S. Hackenbuchner, S. Birner, J. A. Majewski, P. Vogl, and J. J. Finley, "Theory of vertical and lateral Stark shifts of excitons in quantum dots", physica status solidi (c) **0,** 1181-1184 (2003).

[87] F. Holka, P. Neogrády, V. Kellö, M. Urban, and G. H. F. Diercksen, "Polarizabilities of confined two-electron systems: the 2-electron quantum dot, the hydrogen anion, the helium atom and the lithium cation", Molecular Physics **103,** 2747 - 2761 (2005).

[88] S. A. Empedocles and M. G. Bawendi, "Quantum-Confined Stark Effect in Single CdSe Nanocrystallite Quantum Dots", Science **278,** 2114-2117 (1997).

[89] T. Unold, K. Mueller, C. Lienau, T. Elsaesser, and A. D. Wieck, "Optical Stark Effect in a Quantum Dot: Ultrafast Control of Single Exciton Polarizations", Physical Review Letters **92,** 157401-4 (2004).

[90] F. Wang, J. Shan, M. A. Islam, I. P. Herman, M. Bonn, and T. F. Heinz, "Exciton polarizability in semiconductor nanocrystals", Nature Materials **5,** 861-864 (2006).

[91] T. C. Choy, *Effective Medium Theory Principles and Applications* (Oxford University Press, New York, 1999).

[92] C. C. Chen and Y. C. Chou, "Electrical-Conductivity Fluctuations near the Percolation Threshold", Physical Review Letters **54,** 2529 (1985).

[93] E. N. Brothers, G. E. Scuseria, and K. N. Kudin, "Longitudinal Polarizability of Carbon Nanotubes", J. Phys. Chem. B **110,** 12860-12864 (2006).

[94] D. Qi, M. Fischbein, M. Drndic, and S. Selmic, "Efficient polymer-nanocrystal quantum-dot photodetectors", Applied Physics Letters **86,** 093103-3 (2005).

[95] J. Piotrowski and A. Rogalski, "Uncooled long wavelength infrared photon detectors", Infrared Physics & Technology **46,** 115-131 (2004).

[96] M. R. Watts, M. J. Shaw, and G. N. Nielson, "Optical resonators: Microphotonic thermal imaging", Nat Photon **1,** 632-634 (2007).

[97] N. T. Gordon, D. J. Lees, G. Bowen, T. S. Phillips, M. Haigh, C. L. Jones, C. D. Maxey, L. Hipwood, and R. A. Catchpole, "HgCdTe Detectors Operating above 200 K", Journal of Electronic Materials **35,** 1140-1144 (2006).

[98] H. Alause, W. Knap, J. L. Robert, R. Planel, V. Thierry-Mieg, F. H. Julien, K. Zekentes, and V. Mosser, "Room-temperature GaAs/AlGaAs multiple-quantum-well optical modulators for the 3-5 μm atmospheric window", Semiconductor Science and Technology **15,** 724-727 (2000).

[99] P. Bhattacharya, X. H. Su, S. Chakrabarti, G. Ariyawansa, and A. G. U. Perera, "Characteristics of a Tunneling Quantum Dot Infrared Photodetector Operating at Room Temperature", Appl. Phys. Lett. **86,** 191106 (2005).

[100] K. S. Zhuravlev, V. A. Kolosanov, A. G. Milekhin, V. G. Polovinkin, T. S. Shamirzaev, Yu N. Rakov, Yu B. Myakishev, J. Fryar, E. McGlynn, and M. O. Henry, "Infrared light emission from GaAs MESFETs operating at avalanche breakdown conditions", Semicond. Sci. Technol. **19,** S94-S95 (2004).

[101] J. R. Hoff, M. Razeghi, and G. J. Brown, "Effect of the spin split-off band on optical absorption in p-type $Ga_{1-x}In_xAs_yP_{1-y}$ quantum-well infrared detectors", Phys. Rev. B **54,** 10773–10783 (1996).

[102] B. W. Kim, E. Mao, and A. Majerfeld, "Analysis of optical properties of p-type GaAs/AlGaAs superlattices for multiwavelength normal incidence photodetectors", Journal of Applied Physics **81,** 1883-1889 (1997).

[103] S. Ekbote, M. Cahay, and K. Roenker, "Importance of the spin-orbit split-off band on the tunneling properties of holes through $Al_xGa_{1-x}As$/GaAs and InP/$In_yGa_{1-y}As$ heterostructures", Physical Review B **58,** 16315 (1998).

[104] D. G. Esaev, M. B. M. Rinzan, S. G. Matsik, and A. G. U. Perera, "Design and optimization of GaAs/AlGaAs heterojunction infrared detectors", J. Appl. Phys. **96,** 4588-4597 (2004).

[105] D. G. Esaev, M. B. M. Rinzan, S. G. Matsik, A. G. U. Perera, H. C. Liu, B. N. Zvonkov, V. I. Gavrilenko, and A. A. Belyanin, "High performance single emitter homojunction interfacial work function far infrared detectors", J. Appl. Phys. **95,** 512-519 (2004).

[106] A. G. U. Perera, W. Z. Shen, H. C. Liu, M. Buchanan, and W. J. Schaff, "GaAs homojunction interfacial workfunction internal photoemission (HIWIP) far-infrared detectors", Mat. Sci. and Eng. B **74,** 56 (2000).

[107] R. P. G. Karunasiri, J. S. Park, and K. L. Wang, "Normal incidence infrared detector using intervalence-subband transitions in $Si_{1-x}Ge_x$/Si quantum wells", Appl. Phy. Lett. **62,** 2434 (1992).

[108] S. G. Matsik, M. B. M. Rinzan, D. G. Esaev, A. G. U. Perera, H. C. Liu, and M. Buchanan, "20 μm cutoff heterojunction interfacial work function internal photoemission detectors", Appl. Phys. Lett. **84,** 3435-3437 (2004).

[109] H. Jiang and J. Singh, "Strain distribution and electronic spectra of InAs/GaAs self-assembled dots: An eight-band study", Phys Rev B **56,** 4696-4701 (1997).

[110] William, L. Wolfe, George, and J. Zissis, *The Infrared Handbook* (Office of Naval Research, Washington, 1985).

[111] M. E. Levinshtein, S. L. Rumyantsev, and M. Shur, *Handbook series on semiconductor parameters* (World Scientific, Singapore; New Jersey, 1996).

[112] K. Adomi, N. Noto, A. Nakamura, and T. Takenaka, "Characterization of AlGaP/GaP heterostructures grown by MOVPE", Journal of Crystal Growth **124,** 570 (1992).

113 H. Kan, T. Nakamura, H. Katsuno, M. Hagino, and T. Sukegawa, "A highly stable long-life GaP-GaAlP heterojunction cold cathode", Applied Physics Letters **34,** 545-548 (1979).

114 F. Issiki, S. Fukatsu, and Y. Shiraki, "Efficient luminescence from AlP/GaP neighboring confinement structure with AlGaP barrier layers", Applied Physics Letters **67,** 1048 (1995).

115 Y. Nabetani, A. Wakahara, and A. Sasaki, "Photoluminescence properties of AlGaP superlattices", Materials Science and Engineering -Lausanne- B **35,** 454-458 (1995).

116 T. A. Prutskij, V. M. Andreev, and V. R. Larionov, "GaP-AlGaP heterojunction photosensors with photosensitivity maximum at 380-420 nm", Solar Energy Materials and Solar Cells **37,** 349-355 (1995).

117 O. V. Sulima, P. E. Sims, J. A. Cox, M. G. Mauk, R. L. Mueller, R. C. Reedy, Jr., A. M. Khammadov, P. D. Paulson, and G. A. Landis, in *High-temperature AlGaP/GaP solar cells for NASA space missions*,Photovoltaic Energy Conversion, 2003. Proceedings of 3rd World Conference on, 2003, p. 737-740 Vol.1.

118 J. H. Epple, C. Sanchez, T. Chung, K. Y. Cheng, and K. C. Hsieh, "Dry etching of GaP with emphasis on selective etching over AlGaP", Journal of Vacuum Science and Technology B Microelectronics and Nanometer Structures **20,** 2252-2255 (2002).

119 J. Hong, J. W. Lee, C. J. Santana, and C. R. Abernathy, "Comparison of dry etching techniques for InGaP, AlInP and AlGaP", Solid-state electronics. **39,** 1109 (1996).

120 J. Hong, J. W. Lee, C. J. Santana, C. R. Abernathy, S. J. Pearton, W. S. Hobson, and F. Ren, "Plasma etching of InGaP, AlInP and AlGaP in BCl_3 environments", Materials Science and Engineering -Lausanne- B **41,** 247-252 (1996).

[121] S. R. Forrest, P. H. Schmidt, R. B. Wilson, and M. L. Kaplan, "Relationship between the conduction-band discontinuities and band-gap differences of InGaAsP/InP heterojunctions", Applied Physics Letters **45,** 1199-1201 (1984).

[122] E. H. Perea, E. E. Mendez, and C. G. Fonstad, "Electroreflectance of indium gallium arsenide phosphide lattice matched to indium phosphide", Applied Physics Letters **36,** 978-980 (1980).

[123] Y. Yamazoe, T. Nishino, and Y. Hamakawa, "Electroreflectance study of InGaAsP quaternary alloys lattice matched to InP", Quantum Electronics, IEEE Journal of **17,** 139-144 (1981).

[124] S. D. Gunapala, B. F. Levine, D. Ritter, R. A. Hamm, and M. B. Panish, "Lattice-matched InGaAsP/InP Long-Wavelength Quantum Well infrared photodetectors", Appl. Phys. Lett. **60,** 636-638 (1992).

[125] M. Tadic, C. Jelen, S. Slivken, and M. Razeghi, in *Photoresponse of InGaAsP-based p-doped quantum well infrared photodetectors*,Microelectronics, 1997. Proceedings., 1997 21st International Conference on, 1997, p. 315-318 vol.1.

[126] T. Ishikawa and J. E. Bowers, "Band lineup and in-plane effective mass of InGaAsP or InGaAlAs on InP strained-layer quantum well", Quantum Electronics, IEEE Journal of **30,** 562-570 (1994).

[127] T. Sasaki, M. Kitamura, and I. Mito, "Selective metalorganic vapor phase epitaxial growth of InGaAsP / InP layers with bandgap energy control in InGaAs / InGaAsP multiple-quantum well structures", Journal of Crystal Growth **132,** 435-443 (1993).

[128] N. Shamir, D. Ritter, and C. Cytermann, "Beryllium doped InP/InGaAsP heterojunction bipolar transistors", Solid-State Electronics **42,** 2039-2045 (1998).

[129] T. Fukunaga, M. Wada, H. Asano, and T. Hayakawa, "Highly Reliable Operation of High-Power InGaAsP/InGaP/AlGaAs 0.8 μm Separate Confinement Heterostructure Lasers", Jpn. J. Appl. Phys. **34,** L1175-L1177 (1995).

[130] M. Rosenzweig, M. Mohrle, H. Duser, and H. Venghaus, "Threshold-current analysis of InGaAs-InGaAsP multiquantum well separate-confinement lasers", Quantum Electronics, IEEE Journal of **27,** 1804-1811 (1991).

[131] S. G. Wallace, B. J. Robinson, P. Mascher, D. A. Thompson, H. K. Haugen, D. Dalacu, and L. Martinu, in *Optical properties of InGaAsP lattice matched to GaAs*,Lasers and Electro-Optics, 2000. (CLEO 2000). Conference on, 2000, p. 313-314.

[132] K. Wakao, H. Nishi, S. Isozumi, S. Ohsaka, T. Kusunoki, and I. Ushijima, "Low-threshold InGaAsP/InGaP lasers at 810 nm grown on GaAs substrate by LPE", Electronics Letters **20,** 374-375 (1984).

[133] S. G. Wallace, B. J. Robinson, P. Mascher, H. K. Haugen, D. A. Thompson, D. Dalacu, and L. Martinu, "Refractive indices of InGaAsP lattice-matched to GaAs at wavelengths relevant to device design", Applied Physics Letters **76,** 2791-2793 (2000).

[134] J. Bettini, M. M. G. de Carvalho, M. A. Cotta, M. A. A. Pudenzi, N. C. Frateschi, A. Silva Filho, L. P. Cardoso, and R. Landers, "Analysis of Be doping of InGaP lattice matched to GaAs", Journal of Crystal Growth **208,** 65-72 (2000).

[135] O. Ichikawa, N. Fukuhara, M. Hata, T. Nakano, M. Sugiyama, Y. Shimogaki, and Y. Nakano, "High-resolution depth profile of the InGaP-on-GaAs heterointerface by FE-AES and its relationship to device properties", Journal of Crystal Growth **298,** 85-89 (2007).

[136] W.-P. Neo, H. Wang, and K. Radhakrishnan, "Temperature dependence of electron ionization coefficients of InGaP measured in InGaP/GaAs/InGaP DHBT's", Journal of Crystal Growth **268,** 406-409 (2004).

[137] K. Yuan and K. Radhakrishnan, "Growth and characterization of compositionally graded InGaP layers on GaAs substrate by solid-source molecular beam epitaxy", Journal of Crystal Growth **243,** 288-294 (2002).

[138] R. Kúdela, M. Kucera, J. Novák, C. Ferrari, and C. Pelosi, "Study of narrow InGaP/(In)GaAs quantum wells", Journal of Crystal Growth **242,** 132-140 (2002).

[139] T. Nakano, Y. Nakano, and Y. Shimogaki, "Kinetic ellipsometry measurement of InGaP/GaAs hetero-interface formation in MOVPE", Journal of Crystal Growth **221,** 136-141 (2000).

[140] T. K. Sharma, B. M. Arora, M. R. Gokhale, and S. Rajgopalan, "Characterization of InGaP/GaAs heterointerfaces grown by metal organic vapour phase epitaxy", Journal of Crystal Growth **221,** 509-514 (2000).

[141] M. Zorn, T. Trepk, P. Kurpas, M. Weyers, J. T. Zettler, and W. Richter, "In situ monitoring and control of InGaP growth on GaAs in MOVPE", Journal of Crystal Growth **195,** 223-227 (1998).

[142] T. Shitara, K. Eberl, J. Dickmann, and C. Wölk, "Growth of high-quality InGaP and application for modulation-doped structure by molecular beam epitaxy with a GaP source", Journal of Crystal Growth **150,** 1261-1265 (1995).

[143] J. W. Park, D. Pavlidis, S. Mohammadi, J. L. Guyaux, and J. C. Garcia, "Improved Emitter Transit Time Using AlGaAs-GaInP Composite Emitter in GaInP/GaAs Heterojunction Bipolar Transistors", IEEE Transactions on Electron Devices **48,** 1297-1303 (2001).

[144] B. C. Lye, P. A. Houston, H. K. Yow, and C. C. Button, "GaInP/AlGaAs/GaInP Double Heterojunction Bipolar Transistors with Zero Conduction Band Spike at the Collector", IEEE Transactions on Electron Devices ED **45,** 2417-2421 (1998).

[145] J. W. Park, D. Pavlidis, S. Mohammadi, C. Dua, and J. C. Garcia, "Improved High Frequency Performance by Composite Emitter AlGaAs/GaInP Heterojunction Bipolar Transistors Fabricated using Chemical Beam Epitaxy", Conference Series - Institute of Physics **156,** 439-442 (1998).

[146] Z. H. Lu, A. Majerfeld, L. W. Yang, and P. D. Wright, "Effect of carrier recombination at the emitter-base heterojunction on the performance of GaInP/GaAs and AlGaAs heterojunction bipolar transistors", Institute of Physics conference series. **141,** 629 (1995).

[147] F. Bugge, A. Knauer, S. Gramlich, I. Rechenberg, G. Beister, J. Sebastian, H. Wenzel, G. Erbert, and M. Weyers, "MOVPE Growth of AlGaAs/GaInP Diode Lasers", Journal of Electronic Materials **29,** 57-61 (2000).

[148] F. Bugge, C. Beister, G. Erbert, S. Gramlich, K. Vogel, U. Zeimer, and M. Weyers, "GaInP/AlGaAs/GaAs laser diodes with high output power", Institute of Physics conference series. **155,** 573 (1997).

[149] C. R. Huggins, T. A. Cross, and C. M. Hardingham, "Two-Junction Cascade Solar Cells AlGaAs/GaAs and GaInP/GaAs, Performance Data and Industrial Comparison", ESA SP **2,** 543 (1998).

[150] D. Jung, C. A. Parker, J. Ramdani, and S. M. Bedair, "AlGaAs/GaInP heterojunction tunnel diode for cascade solar cell application", Journal of Applied Physics **74,** 2090 (1993).

151 "Growth and development of GaInAsP for use in high-efficiency solar cells. Annual subcontract report, 1 July 1991--30 June 1992", http://www.osti.gov/servlets/purl/10143733-w4cBq0/webviewable/,

152 R. C. Jayasinghe, G. Ariyawansa, N. Dietz, A. G. U. Perera, S. G. Matsik, H. B. Yu, I. T. Ferguson, A. Bezinger, S. R. Laframboise, M. Buchanan, and H. C. Liu, "Simultaneous detection of ultraviolet and infrared radiation in a single GaN/GaAlN heterojunction", Optics Letters **33,** In Press (2008).

153 B. K. Ng, J. P. R. David, R. C. Tozer, M. Hopkinson, G. Hill, and G. J. Rees, "Photodetectors and Photoreceivers - Excess Noise Characteristics of $Al_{0.8}Ga_{0.2}As$ Avalanche Photodiodes", IEEE photonics technology letters : a publication of the IEEE Laser and Electro-optics Society. **14,** 522 (2002).

154 B. Guan, X. Guo, T. Liang, X. Gu, J. Deng, J. Guo, H. Yang, Q. Lin, and G. Shen, "Sacrificial $Al_{0.8}Ga_{0.2}As$ etching for microstructures in integrated optoelectronic devices", Journal of applied physics. **100,** 113508 (2006).

155 A. R. Mirabedini, L. J. Mawst, D. Botez, and R. A. Marsland, "High peak-current-density strained-layer $In_{0.3}Ga_{0.7}As$/ $Al_{0.8}Ga_{0.2}As$ resonant tunneling diodes grown by metal-organic chemical vapor deposition", Applied physics letters. **70,** 2867 (1997).

156 B. K. Ng, J. P. R. David, S. A. Plimmer, G. J. Rees, R. C. Tozer, M. Hopkinson, and G. Hill, "Avalanche Multiplication Characteristics of $Al_{0.8}Ga_{0.2}As$ Diodes", IEEE transactions on electron devices. **48,** 2198 (2001).

157 B. K. Ng, J. P. R. David, S. A. Plimmer, M. Hopkinson, R. C. Tozer, and G. J. Rees, "Impact ionization coefficients of $Al_{0.8}Ga_{0.2}As$", Applied physics letters. **77,** 4374 (2000).

[158] F. Echeverria, P. Skeldon, G. E. Thompson, M. J. Graham, H. Habazaki, and K. Shimizu, "Formation of anodic oxides on $Al_{0.2}Ga_{0.8}As$ and $Al_{0.8}Ga_{0.2}As$ in tungstate electrolytes", Corrosion science. **42,** 1839-1851 (2000).

[159] D. M. Schaadt, B. Feng, and E. T. Yu, "Enhanced semiconductor optical absorption via surface plasmon excitation in metal nanoparticles", Applied Physics Letters **86,** 063106-3 (2005).

[160] A. Rogalski, "Optical detectors for focal plane arrays", Opto-Electronics Review **12,** 221-245 (2004).

[161] M. Z. Tidrow, W. W. Clark Iii, W. Tipton, R. Hoffman, W. A. Beck, S. C. Tidrow, D. N. Robertson, H. K. Pollehn, K. R. Udayakumar, H. R. Beratan, K. L. Soch, C. M. Hanson, and M. Wigdor, in *Uncooled infrared detectors and focal plane arrays*,Detectors, Focal Plane Arrays, and Imaging Devices II, Beijing, China, 1998 (SPIE), p. 178-187.

[162] S. W. Smye, J. M. Chamberlain, A. J. Fitzgerald, and E. Berry, "The interaction between Terahertz radiation and biological tissue", Physics in Medicine and Biology **46,** R101-R112 (2001).

[163] D. Kindl, J. Touskova, E. Hulicius, J. Pangrac, T. Simecek, V. Jurka, P. Hubik, J. J. Mares, and J. Kristofik, "Influence of growth rate on charge transport in GaSb homojunctions prepared by metalorganic vapor phase epitaxy", Journal of Applied Physics **95,** 1811-1815 (2004).

[164] Z. G. Hu, A. G. U. Perera, Y. Paltiel, A. Raizman, and A. Sher, "Zn-doped GaSb epitaxial film absorption coefficients at terahertz frequencies and detector applications", J. Appl. Phys. **98,** 023511-5 (2005).

[165] M. Razeghi and H. Mosheni, in *Handbook of infrared detection technologies*; *Vol. 1*, 1 ed., edited by M. Henini and M. Razeghi (Elsevier, UK, 2002), p. 191–232.

[166] J. B. Rodriguez, P. Christol, F. Chevrier, and A. Joullie, "Optical characterization of symmetric InAs/GaSb superlattices for detection in the 3-5 [mu]m spectral region", Physica E: Low-dimensional Systems and Nanostructures **28,** 128-133 (2005).

[167] H. Mosheni, J. Wojkowski, M. Razeghi, G. Brown, and W. Mitchel, "Uncooled InAs/GaSb type-II infrared detectors grown on GaAs substrates for the 8-12 μm atmospheric windows", IEEE J. Quantum Electron. Lett. **35,** 1041-1044 (1999).

[168] H. Mohseni, M. Razeghi, G. J. Brown, and Y. S. Park, "High-performance InAs/GaSb superlattice photodiodes for the very long wavelength infrared range", Applied Physics Letters **78,** 2107-2109 (2001).

[169] Y. Paltiel, A. Zussman, N. Snapi, A. Sher, G. Jung, K. Cohen, E. Benory, and E. Weiss, "Voltage tunability of high performance Zn doped p-type QWIP grown by MOVPE", Infrared Physics & Technology **47,** 37-42 (2005).

[170] W. Z. Shen and A. G. U. Perera, "Photoconductivity in homojunction internal photoemission far-infrared detectors", Infrared Physics and Technology **39,** 329 (1998).

[171] K. Seeger, *Semiconductor Physics (fifth edition)* (Springer-Verlag, ch.6, 1991).

[172] Y.-K. Su, Y.-Z. Chiou, F.-S. Juang, S.-J. Chang, and J.-K. Sheu, "GaN and InGaN Metal-Semiconductor-Metal Photodetectors with Different Schottky Contact Metals", Jpn. J. Appl. Phys. **40,** 2996-2999 (2001).

[173] A. G. U. Perera, in *Homojunction and Quantum-Well Infrared Detectors*; *Vol. 21*, edited by M. H. Francombe and J. L. Vossen (edited by M. H. Francombe and J. L. Vossen Academic Press, NY, 1995), p. 1-75.

174 D. G. Esaev, S. G. Matsik, M. B. M. Rinzan, A. G. U. Perera, H. C. Liu, and M. Buchanan, "Resonant cavity enhancement in heterojunction GaAs/AlGaAs terahertz detectors", J. Appl. Phys. **93,** 1879-1883 (2003).

Appendix A

Detector characterization

A.1 Responsivity (Photo Sensitivity)

Responsivity is the output current (or output voltage) per watt of incident power when noise is not a consideration.

$$R = \frac{S}{PA} \tag{A.1}$$

where S is the signal output [A or V], P is the incident light intensity [W/cm^2], and A is the detector active area [in cm^2].

A.1.1 Spectral response measurements and responsivity calibration

The samples responding in the UV, VIS or NIR range was measured using a monochromator and lock-in-amplifier setup (see details in Appendix B). A Si photodiode and an InGaAs photodiode with a known responsivity curves were used to calculate the spectral responsivity of the sample. The samples responding in the MIR and FIR range were measured using a Perkin Elmer System 2000 Fourier Transform Infrared (FTIR) Spectrometer and a Si bolometer was used to calculate responsivity spectra. The measurement procedure involves two measurements: sample spectra

($V_S(\lambda)$) and bolometer or calibrated Si or InGaAs photodiode spectra ($V_{CD}(\lambda)$). Those two spectra are obtained concurrently with the same combination of optical windows, beamsplitter, and filters, so that the optical path is identical. The voltage responsivity can be obtained using

$$R_V(\lambda) = \frac{GV_S(\lambda)}{V_{CD}(\lambda)} R_{CD}(\lambda)\,. \tag{A.2}$$

Here G is the geometrical factor, which corrects for differences in the radiation-incident-area of the detector and the bolometer/photodiode and R_{CD} is the sensitivity curve given by the manufacturer. To obtain the current responsivity, the voltage responsivity is divided by the effective resistance. As the detector and the load resistor act as a voltage divider for the photocurrent, the effective resistance R_e is the parallel combination of the load R_L and the detector dynamic resistance R_d = dV/dI, yielding $R_e = R_L \times R_d /(R_L + R_d)$. Therefore, the current responsivity can be express as

$$R_I(\lambda) = \frac{GV_S(\lambda) R_{CD}(\lambda)}{V_{CD}(\lambda)} \times \frac{(R_L + R_d)}{R_L R_d} \tag{A.3}$$

A.2 Quantum efficiency (η)

The photocurrent of the detector for given wavelength λ can be expressed as

$$I_{SC} = \eta q g \frac{PA}{h\frac{c}{\lambda}} = \frac{\eta q g P A \lambda}{hc} \tag{A.4}$$

where λ is the wavelength, h is Planck's constant, c is the speed of light and q is the electron charge. The quantum efficiency η is the number of electron hole pairs generated per incident photon, and the gain, g, represents the number of carriers collected at the contact per one generated carrier.

Thus, the current responsivity is

$$R_I(\lambda) = \frac{I_{SC}}{PA} = \frac{\eta q g \lambda}{hc} = \frac{\eta \lambda g}{1.24} \tag{A.5}$$

if λ is measured in μm and R_I in A/W. The relation with quantum efficiency η is as follows:

$$\eta = 1.24 \frac{R_I(\lambda)}{\lambda g} \times 100\%. \tag{A.6}$$

A.3 Noise Equivalent Power (NEP)

The incident light power needed to produce an output signal equal to the intrinsic noise level of the detector is

$$\mathrm{NEP} = \frac{PA}{S/N.\sqrt{\Delta f}} \tag{A.7}$$

where N is the noise output and Δf is the noise band width.

A.4 Specific Detectivity (D^*)

The specific detectivity, a figure of merit used to characterize performance of a detector, is equal to the reciprocal of noise equivalent power, normalized to unit area and unit bandwidth and is given by

$$D^* = \frac{S/N.\sqrt{\Delta f}}{P\sqrt{A}} = \frac{\sqrt{A}}{\mathrm{NEP}} = \frac{R_I.\sqrt{A}}{N/\sqrt{\Delta f}}. \tag{A.8}$$

The noise current density N_d can be measured using a noise spectrum analyzer and is expressed as

$$N_d = \frac{N^2}{\Delta f}. \tag{A.9}$$

Thus D* can be calculated using following formula

$$D^* = \frac{R_I\sqrt{A}}{\sqrt{N_d}} \qquad [\mathrm{cm\ Hz^{1/2}/W}]. \tag{A.10}$$

In general, the measurement conditions of D^* are expressed in the format of D^*(a, b, c), where "a" is the temperature [K] or wavelength [µm] of a radiant source, "b" is the chopping frequency, and "c" is the bandwidth.

A.4.1 Noise measurements and detectivity calculation

The D^* of the devices at different temperatures and applied biases is obtained from the measured responsivity (R_I), noise current density (N_d), and the illuminated area of the detector (A). The later are measured with a dual channel Fast Fourier Transform (FFT) signal analyzer and a SR570 low noise current pre-amplifier, as shown in Figure B.15. A thick copper plate at the device temperature is used as the radiation block to provide the dark conditions for the measurements. The value of D^* is calculated from the above equation.

Appendix B

Software development with Microsoft Visual Basic

B.1 Multi-sample I-V-T characterization software

An in-house software program was developed to automate the detector I-V-T (current vs. voltage under different temperatures) characterization process. The software controls a Keithley 2400 source meter, a Keithley 7001 switch sysytem, a SI-9620 temperature controller (for a Helium closed-cycle refrigerator) through the GPIB interface. The system is capable of recording I-V curves of up to 10 samples in a user defined voltage range, temperature range, and step size. In general, a closed–cycle refrigerator is capable of working from 7 K up to 330 K in temperature. User selectable options allow the user to save data directly in a Microsoft Excel file or ASCII text files. As a additional feature the user can enable current vs. time data recording at a fixed voltage, for studying transient characteristics of the sample. The following figures show the user interfaces and main program windows. Source codes for each form and subprograms are listed in the next subsection.

B.1.1 I-V-T setup instrument configuration

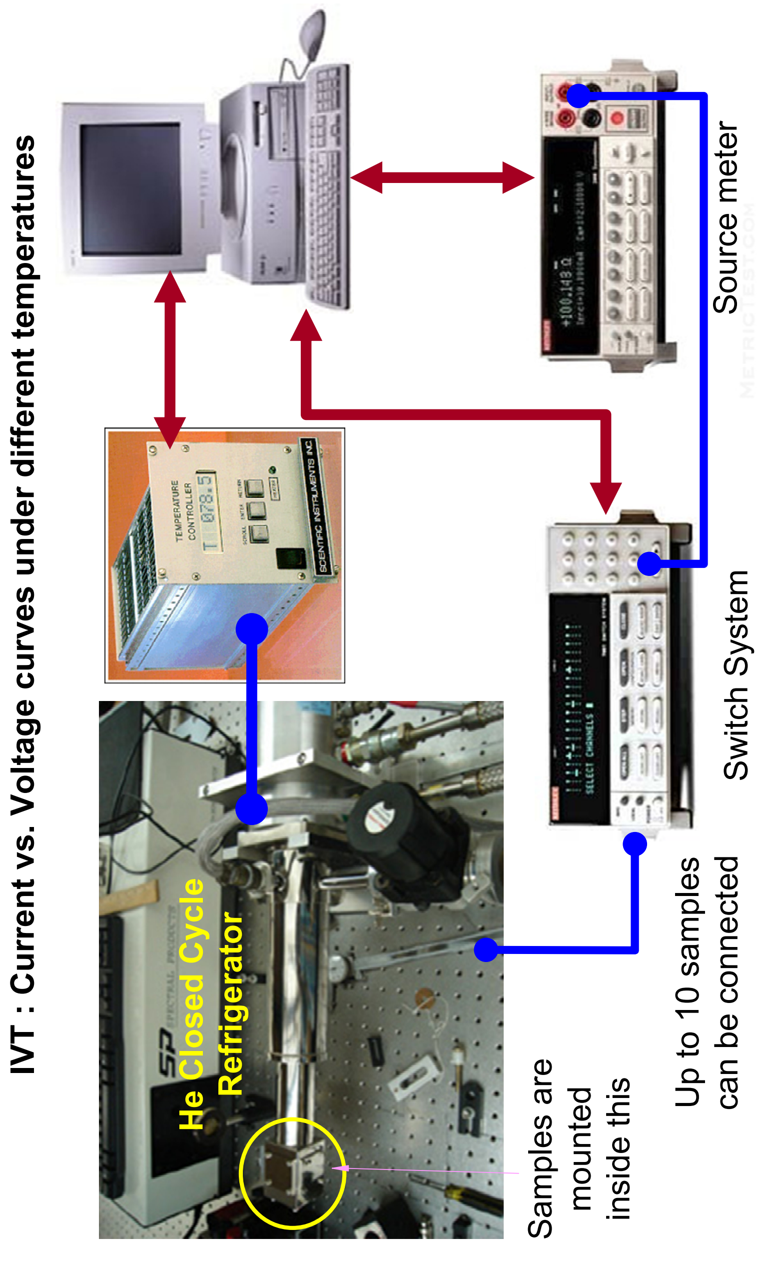


Figure B.1 The block diagram of the I-V-T setup showing GPIB communication connections (red lines with arrowheads) and other sample wirings (blue lines).

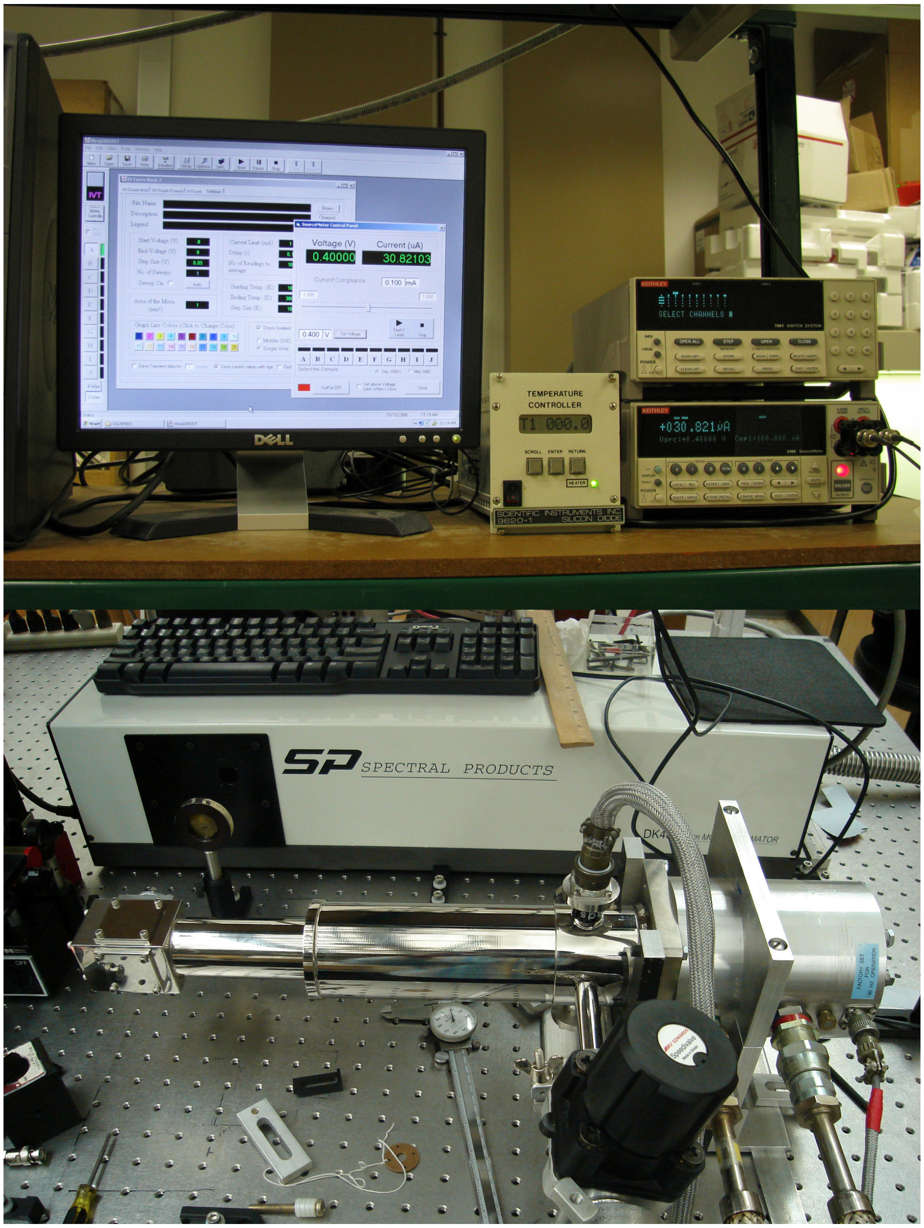


Figure B.2 Photograph of the IVT setup showing Keithley 2400 source meter, 7001 switch system, SI-9620 temperature controller (top) and helium closed cycle refrigerator (bottom).

B.1.2 User interfaces of the I-V-T software

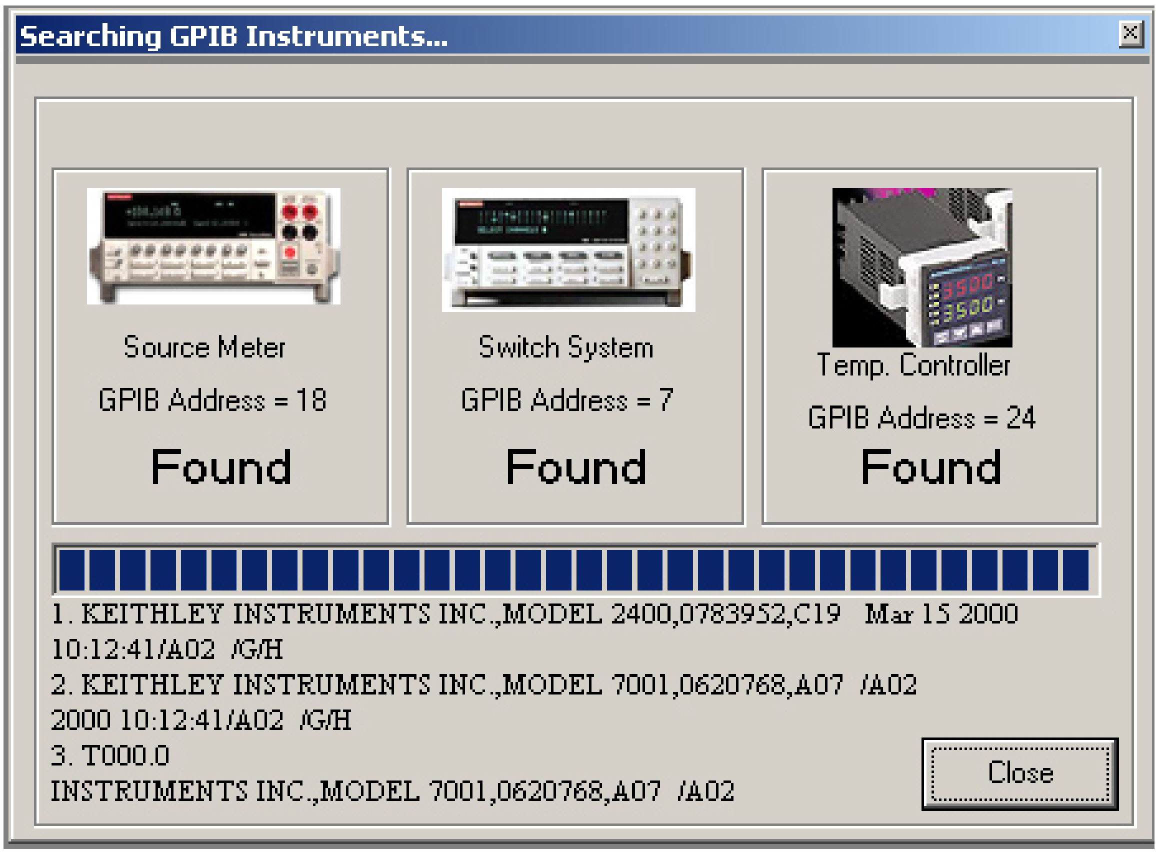


Figure B.3 The initializing window searches communication status of all three instruments in order to initialize GPIB communication. If a communication error occurs or instrument is not physically connected, it will inform the user to check the particular instrument and connections.

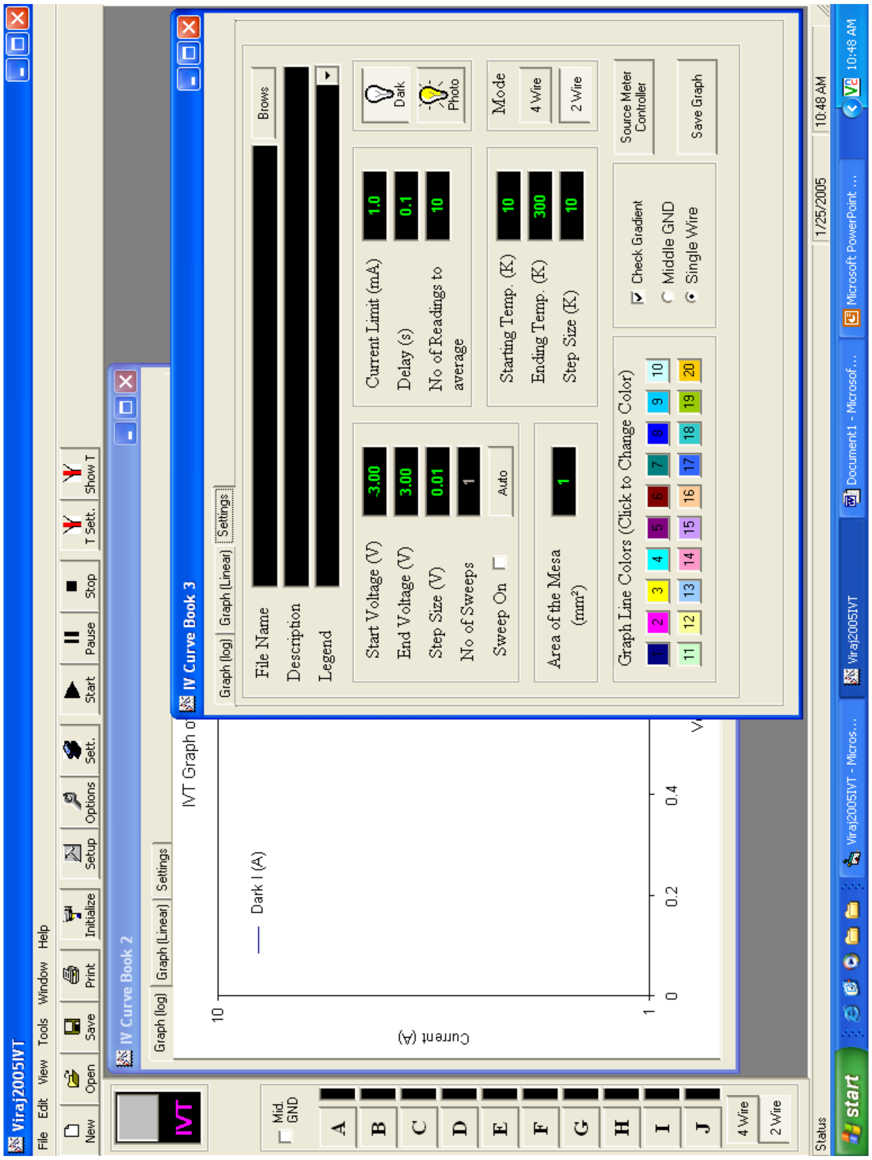


Figure B.4 The main program window showing manual I-V mode operation settings. In this mode user can select the samples by clicking the buttons labeled A to J and enter scan parameters and a temperature. The “Settings” tab and “Graph (log)” tab views are shown in the figure.

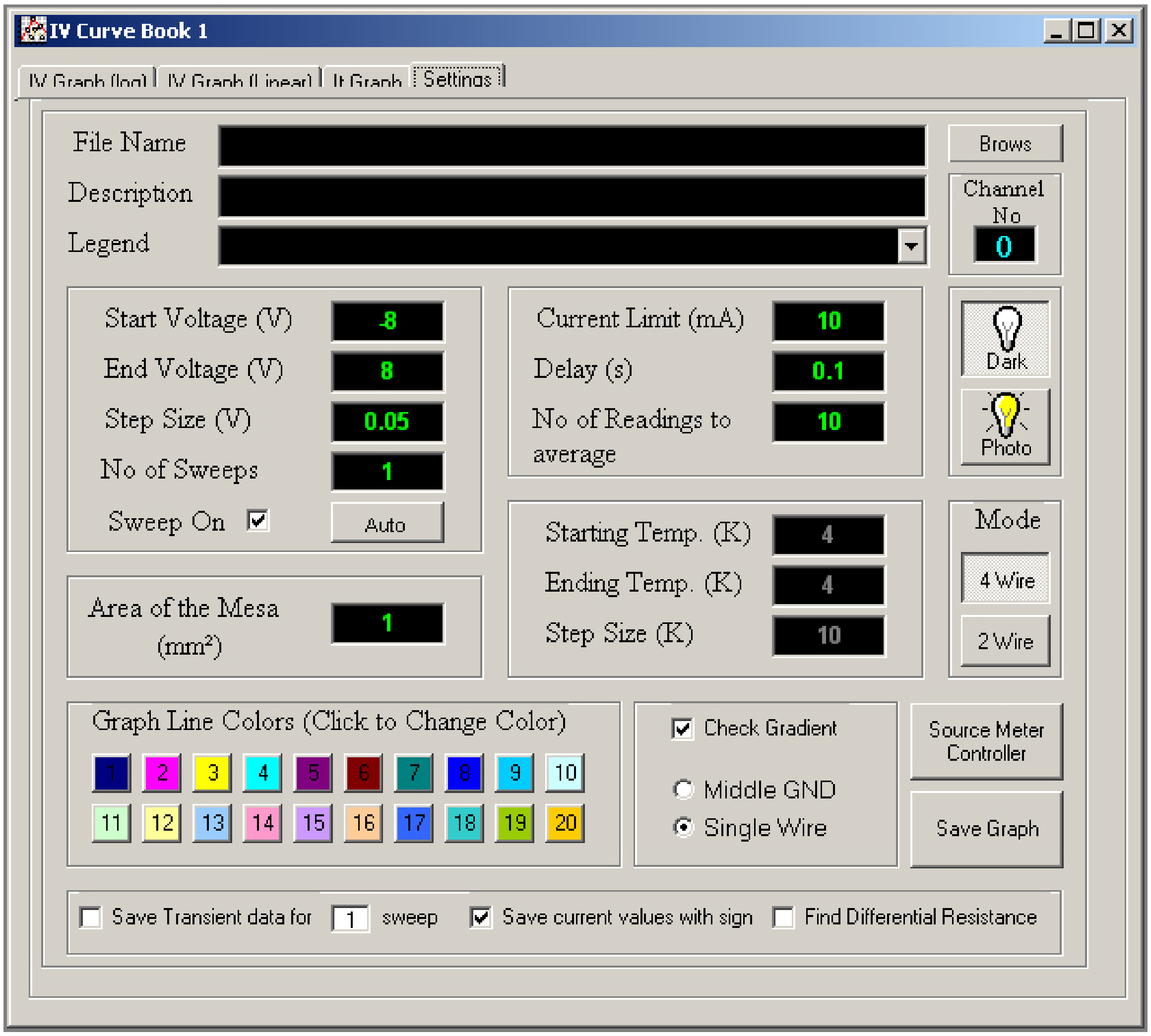


Figure B.5 The user interface for the I-V mode settings. By clicking different tabs in the top of the window the user can view different graphs such as the I-V plot on log scale, I-V plots on a linear scale, or an I-t plot for each voltage.

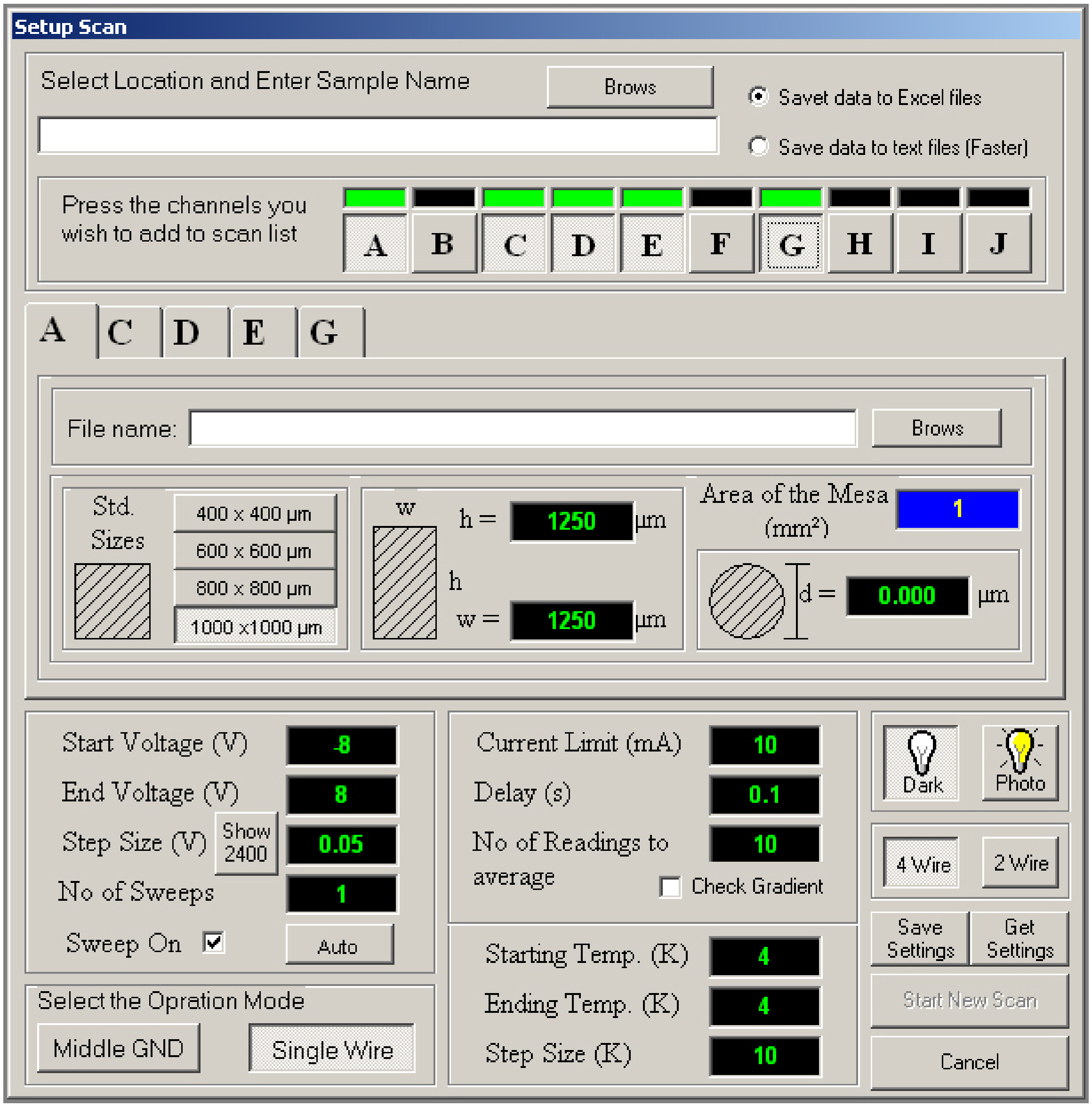


Figure B.6 User interface for the multi sample I-V-T scan setup. In this mode user can select the samples to be tested (labeled as A-J), voltage range, temperature range… etc. When a scan is started, the program will set the starting temperature. After it reaches the specified temperature, it will switch samples one at a time and record I-V curves. This process is repeated at each specified temperature.

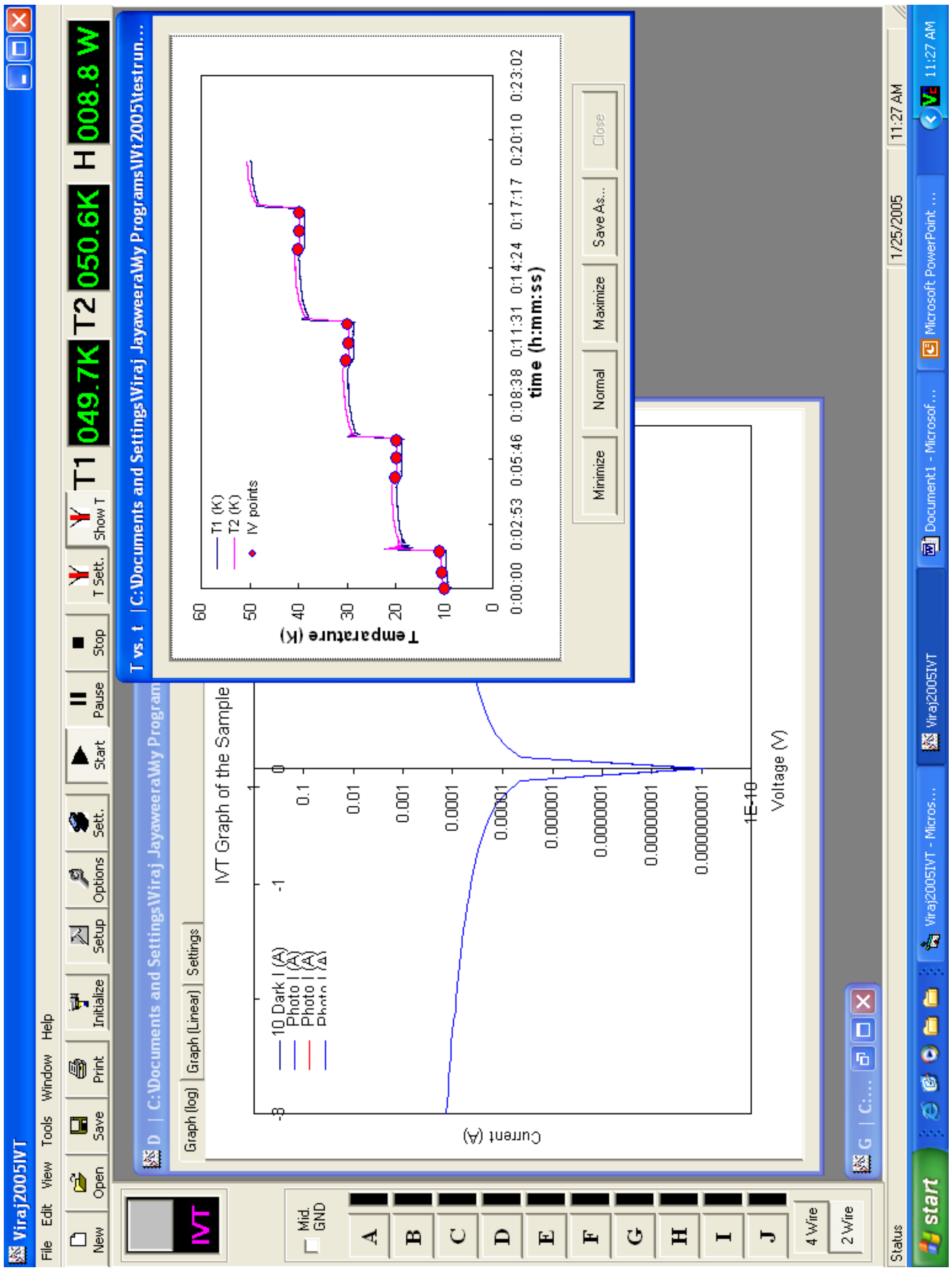


Figure B.7 The I-V-T mode main screen showing the active sample graph in the left window and the sample temperature variation with the time in the right window.

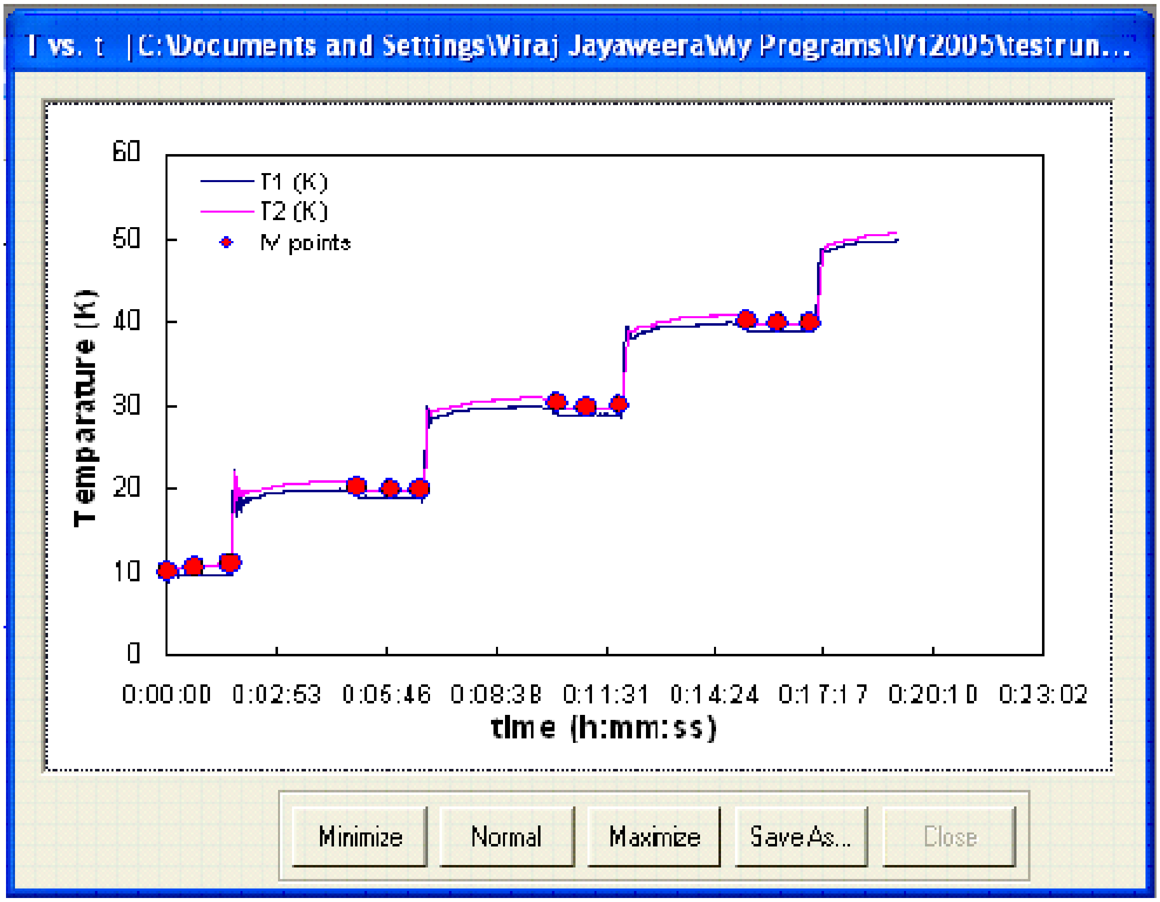


Figure B.8 The temperature vs. time plot for the I-V-T mode. Red circles indicate the measurement starting point.

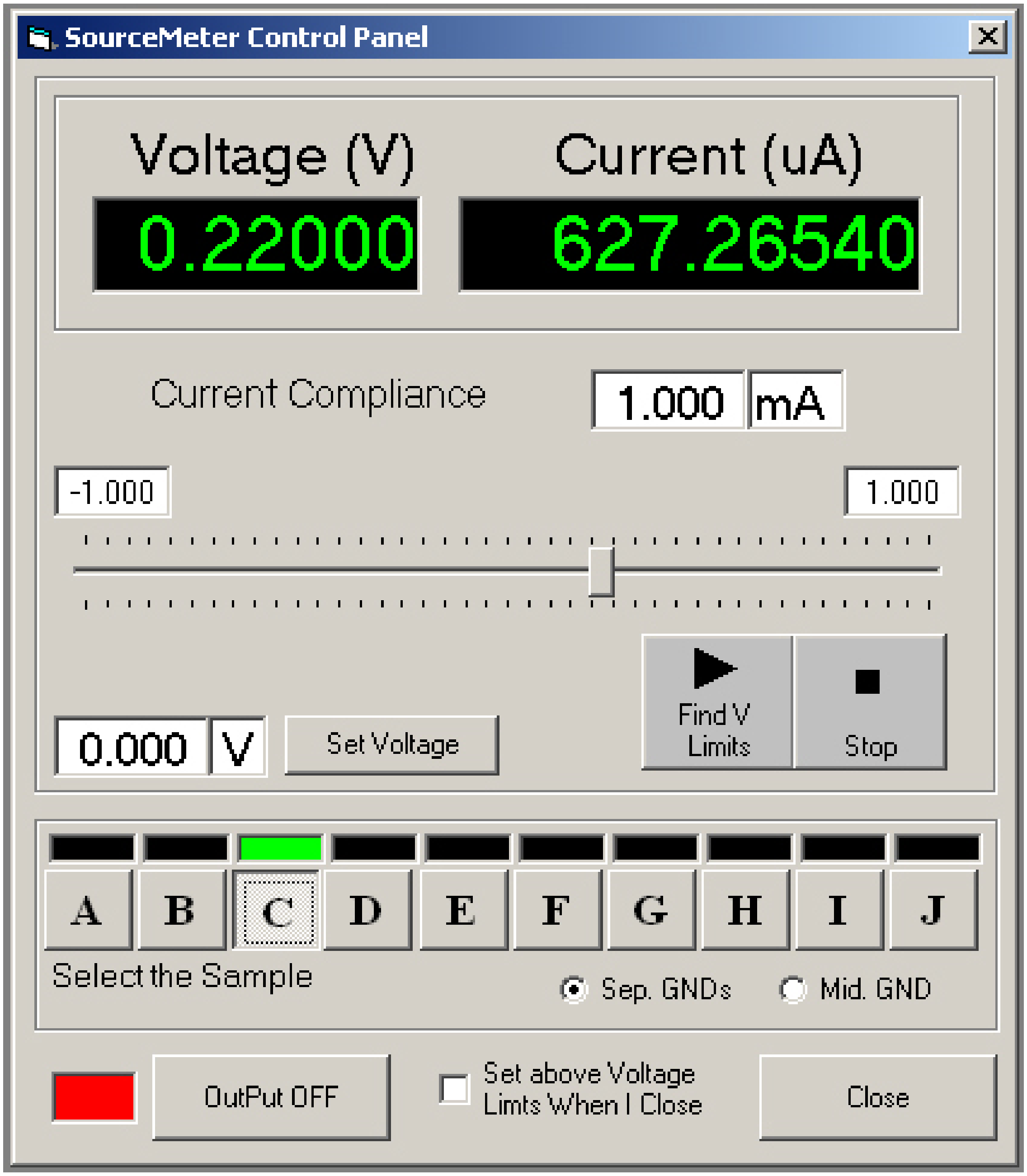


Figure B.9 The control panel for the Keithley 2400 source meter. The user can use this to find scanning voltage limits based on a specified current level for their samples.

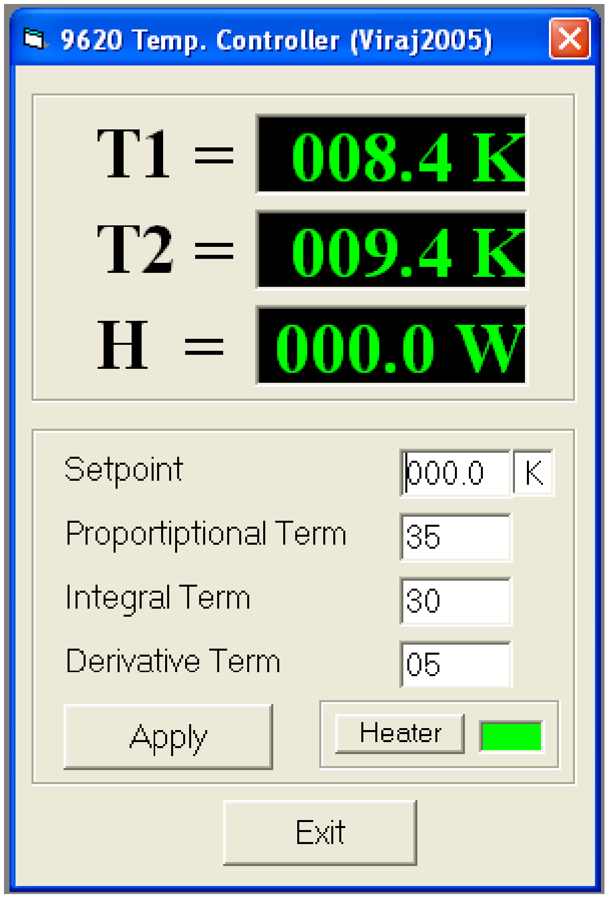


Figure B.10 The control panel for the SI9620 temperature controller. This can be used to observe live temperatures close to the sample (T2), close to the heating element (T1) and the actual heater power (H). The user also can set the temperature to a specific value by entering a value to "setpoint". Three constants (proportional term, integral term, and derivative term) can also be changed that the temperature controller algorithm needs to determine the heater power.

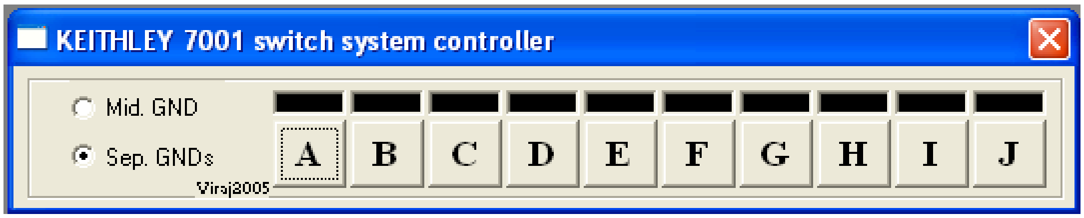
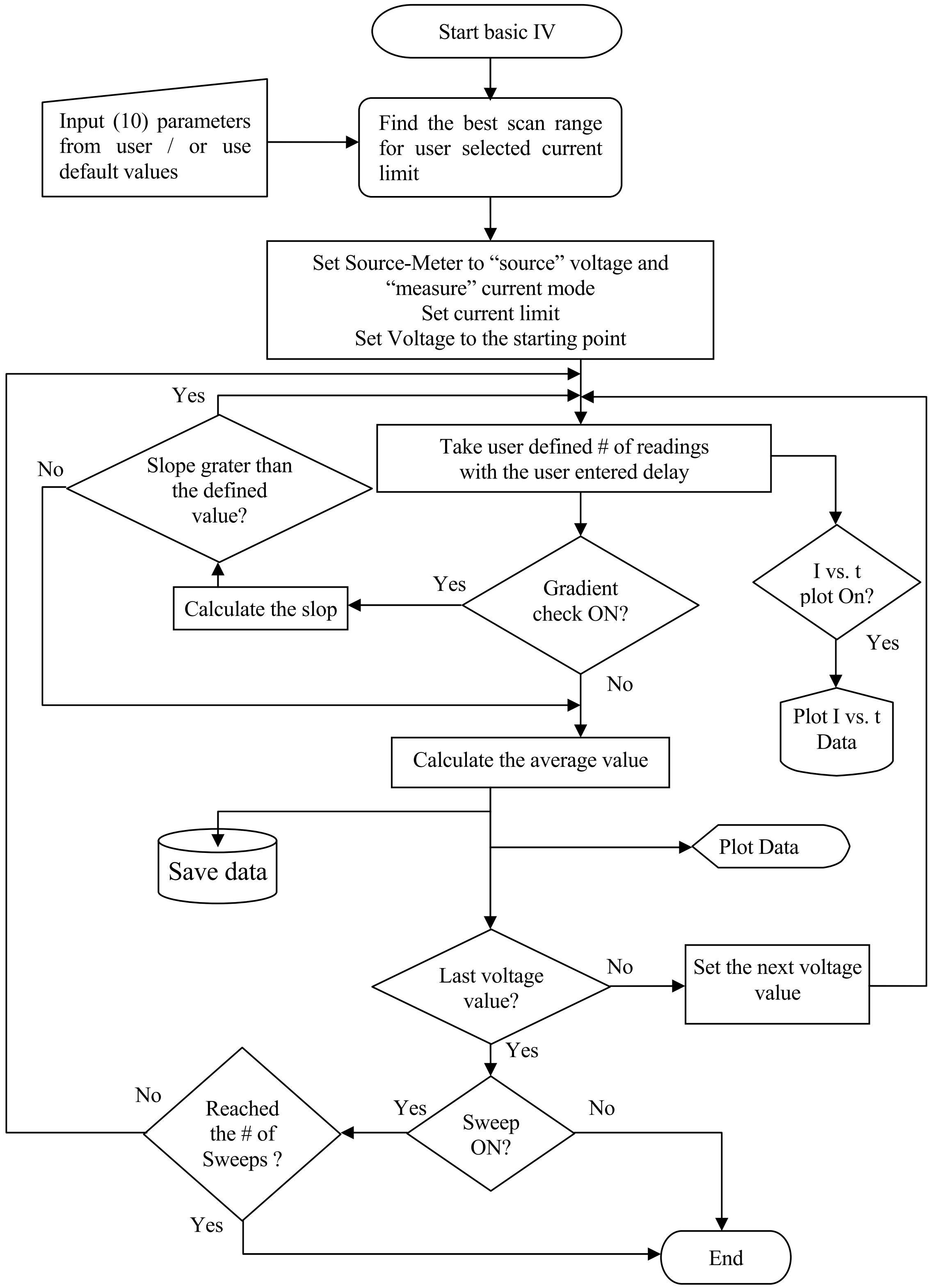


Figure B.11 This panel controls the Keithley 7001 switch system and can be used as standalone software. By clicking corresponding button, the user can connect the corresponding sample (A – J) to a measuring instrument (source meter, volt meter, preamplifier... etc.). The switch system is wired in four wire configurations to allow the user a to use wide variety of instruments which have four inputs (two wires to apply current and two for measuring voltage) or two inputs.

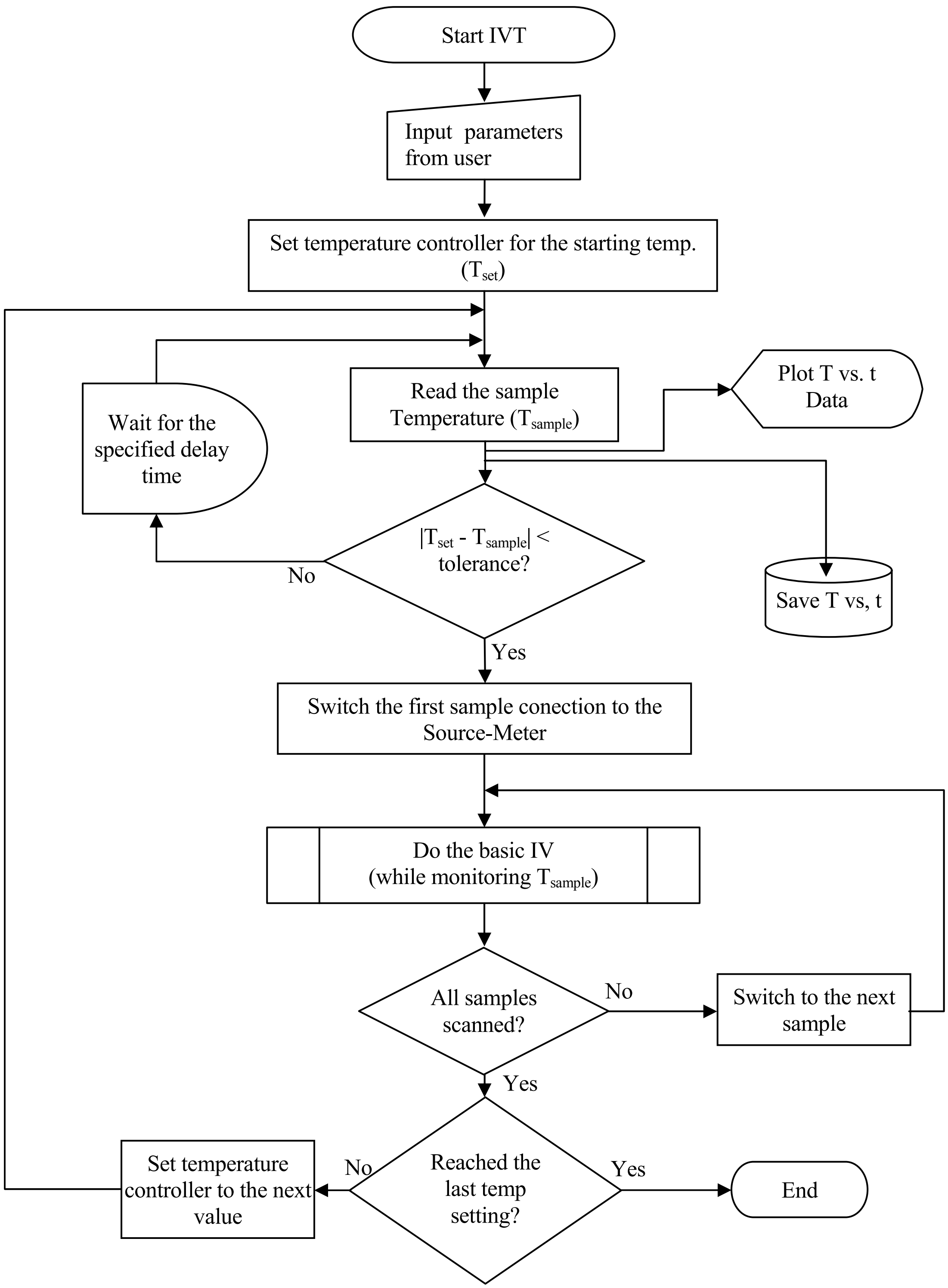
B.1.3 Flow chart for the IV opration mode

(Kethley 2004 source meter, 7001 switch system)



B.1.4 Flow chart for the multi sample IVT operation mode

(Kethley 2004 source meter, 7001 switch system)



B.1.5 Source code

```
************************************************************************
Module1.bas
************************************************************************
'Copyrights – Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State
University

Public fMainForm As frmMain
Public IVTMode As Boolean 'If current running mode is IVT then IVTMode is true

Sub main()
    InitializeSM
    InitializeSW
    InitializeTC
    frmSplash.Refresh
    Set fMainForm = New frmMain
    Load fMainForm
    Unload frmSplash
    fMainForm.Show
    frmMessage.Show
End Sub

Sub SaveSetings()
    fn = FreeFile()
    Open "IVtSettings.inf" For Output As #fn
    Print #fn, GPIBSM
End Sub

Sub Wait(Time_in_seconds As Single)
    st = Timer
    While Timer - st < Time_in_seconds
        DoEvents
    Wend
End Sub


************************************************************************
GPIB_commands.bas
************************************************************************
'Copyrights – Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State
University

Const ARRAYSIZE = 1024             ' Size of read buffer
Dim ErrMsg As String * 100
Dim ErrorMnemonic
Dim ValueStr As String * ARRAYSIZE
Dim Response As Integer
Dim DevSW%
Dim DevSM%
Dim DevTC%
Public GPIBSM As Integer ' source meter GPIB address
Public GPIBSW As Integer ' Switch GPIB address
Public GPIBTC As Integer ' Temp. Controller GPIB address
```

```
'To write command to KEITHLEY 2400 Source Meter
Sub SendSM(CommandString, stat%)
    ilwrt DevSM%, CommandString, Len(CommandString)
    If (ibsta And EERR) Then
        stat% = 1
        Call  GPIBCleanup("Unable  to  write  to  KEITHLEY  2400  SourceMeter",
DevSM%)
    Else
        stat% = 0
    End If
End Sub

'To write command to KEITHLEY 7001 Switch system
Sub SendSW(CommandString, stat%)
    ilwrt DevSW%, CommandString, Len(CommandString)
    If (ibsta And EERR) Then
        stat% = 1
        Call  GPIBCleanup("Unable  to  write  to  KEITHLEY  7001  Switch  system",
DevSW%)
    Else
        stat% = 0
    End If
End Sub
'To write command to SI 9620-1 Temperature controler
Sub SendTC(CommandString, stat%)
    ilwrt DevTC%, CommandString, Len(CommandString)
    If (ibsta And EERR) Then
        stat% = 1
        Call GPIBCleanup("Unable to write to Temp. Controller", DevTC%)
    Else
        stat% = 0
    End If
End Sub

'To read Data from KEITHLEY 2400 Source Meter
Sub ReadSM(r$)
    ilrd DevSM%, ValueStr, Len(ValueStr)
    If (ibsta And EERR) Then
        Call GPIBCleanup("Unable to read from device", DevSM%)
    End If
    r$ = ValueStr
End Sub

'To read Data from 7001 Switch system
Sub ReadSW(r$)
    ilrd DevSW%, ValueStr, Len(ValueStr)
    If (ibsta And EERR) Then
        Call GPIBCleanup("Unable to read from device", DevSW%)
    End If
    r$ = ValueStr
End Sub

'To Queries Temp. Controller SI 9620-1 settings
Function ReadTC(CmdString As String)
    ilwrt DevTC%, CmdString, Len(CmdString)
    If (ibsta And EERR) Then
        Call GPIBCleanup("Unable to write to device", DevTC%)
```

```
    End If
    ilrd DevTC%, ValueStr, Len(ValueStr)
    If (ibsta And EERR) Then
        Call GPIBCleanup("Unable to read from device", DevTC%)
    End If
    ReadTC = ValueStr
End Function

' To close required channel
Sub CloseSW(j As Integer, MidGNDMode As Boolean)
    SendSW ":open all", stat%
    If j < 1 Or j > 10 Then Exit Sub
    cl$ = ":clos (@ 1!" & Trim(Str(j)) & ", 1!" & Trim(Str(j + 10)) & ")"
    SendSW cl$, stat%
    If MidGNDMode = True Then
        SendSW ":clos (@ 1!22)", stat%
    Else
        SendSW ":clos (@ 1!21)", stat%
    End If
End Sub
Sub InitializeSM()
    GPIBSM = GetSetting(App.Title, "Settings", "GPIBSM", 18)
    DevSM% = ildev(0, GPIBSM, 0, T10s, 1, 0)
    If (ibsta And EERR) Then
        ErrMsg = "Unable to open KEITHLEY 2400 Source Meter" & Chr(13) & "ibsta
= &H" & _
        Chr(13) & Hex(ibsta) & "iberr = " & iberr
        MsgBox ErrMsg, vbCritical, "Error"
        Exit Sub
    End If
    ilclr DevSM%
End Sub

Sub InitializeSW()
    GPIBSW = GetSetting(App.Title, "Settings", "GPIBSW", 7)
    DevSW% = ildev(0, GPIBSW, 0, T10s, 1, 0)
    If (ibsta And EERR) Then
        ErrMsg = "Unable to open KEITHLEY 7001 Switch system" & Chr(13) &
"ibsta = &H" & _
        Chr(13) & Hex(ibsta) & "iberr = " & iberr
        MsgBox ErrMsg, vbCritical, "Error"
        Exit Sub
    End If
    ilclr DevSW%
End Sub

Sub InitializeTC()
    GPIBTC = GetSetting(App.Title, "Settings", "GPIBTC", 24)
    DevTC% = ildev(0, GPIBTC, 0, T10s, 1, 0)
    If (ibsta And EERR) Then
        ErrMsg = "Unable to open Temp. Controller" & Chr(13) & "ibsta = &H" & _
        Chr(13) & Hex(ibsta) & "iberr = " & iberr
        MsgBox ErrMsg, vbCritical, "Error"
        Exit Sub
    End If
    ilclr DevTC%
End Sub
```

```
Sub GPIBCleanup(msg$, Dev%)

    ' After each GPIB call, the application checks whether the call
    ' succeeded. If an NI-488.2 call fails, the GPIB driver sets the
    ' corresponding bit in the global status variable. If the call
    ' failed, this procedure prints an error message, takes the device
    ' offline and exits.

    'ErrorMnemonic = Array("EDVR", "ECIC", "ENOL", "EADR", "EARG", _
                      "ESAC", "EABO", "ENEB", "EDMA", "", _
                      "EOIP", "ECAP", "EFSO", "", "EBUS", _
                      "ESTB", "ESRQ", "", "", "", "ETAB")

    'ErrMsg$ = msg$ & Chr(13) & "ibsta = &H" & Hex(ibsta) & Chr(13) _
              & "iberr = " & iberr & " <" & ErrorMnemonic(iberr) & ">"
    'MsgBox ErrMsg$, vbCritical, "Error"
    ilonl Dev%, 0

End Sub

Sub FindInstruments()
    InitializeSM
    InitializeSW
    InitializeTC



End Sub
```

****

frmMessage.frm

****

```
'Copyrights – Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State
University


Private Sub cmdClose_Click()
    Unload Me
End Sub

Private Sub Form_Load()
    Me.Left = 3465
    Me.Top = 1800
    Me.Visible = True
    Me.pgbSearch.value = 5
    DoEvents
    InitializeSM
    InitializeSW
    InitializeTC
    Me.lblGPIBSM.Caption = "GPIB Address = " & GPIBSM
    Me.lblGPIBSW.Caption = "GPIB Address = " & GPIBSW
    Me.lblGPIBTC.Caption = "GPIB Address = " & GPIBTC
    For i = 5 To 33
```

```
        Me.pgbSearch.value = i
        DoEvents
        Wait 0.005
    Next i
    SendSM "*IDN?", stat%
    If stat% = 0 Then
        ReadSM r$
        Me.lblInstrment.Caption = "1. " & r$
        Me.lblStatusSM.Caption = "Found"
        SendSM ":SYST:RSEN OFF", stat%
        DoEvents
    Else
        Me.lblStatusSM.Caption = "Not Found"
    End If

  '********************************************
    SendSW "*IDN?", stat%
    For i = 33 To 66
        Me.pgbSearch.value = i
        DoEvents
        Wait 0.005
    Next i
    If stat% = 0 Then
        ReadSW r$
        Me.lblInstrment.Caption = lblInstrment.Caption & Chr(13) & "2. " & r$

        Me.lblStatusSW.Caption = "Found"
        fMainForm.chkMidGND.Enabled = True
        For i = 0 To 9
            fMainForm.TgB(i).Enabled = True
        Next i
    Else
        Me.lblStatusSW.Caption = "Not Found"
        fMainForm.chkMidGND.Enabled = False
        For i = 0 To 9
            fMainForm.TgB(i).Enabled = False
        Next i
    End If
  '********************************************
    SendTC "*IDN?", stat%
    For i = 66 To 100
        Me.pgbSearch.value = i
        DoEvents
        Wait 0.005
    Next i
    If stat% = 0 Then
        Me.lblInstrment.Caption  =  lblInstrment.Caption  &  Chr(13)  &  "3.  "  &
ReadTC("*IDN?" & Chr(13))
        Me.lblStatusTC.Caption = "Found"
        fMainForm.tbToolBar.Buttons.Item(16).Enabled = True
        fMainForm.tbToolBar.Buttons.Item(17).Enabled = True
    Else
        Me.lblStatusTC.Caption = "Not Found"
        fMainForm.tbToolBar.Buttons.Item(16).Enabled = False
        fMainForm.tbToolBar.Buttons.Item(17).Enabled = False
    End If
    Me.lblSearching.Visible = False
```

```
    Me.pgbSearch.value = 100
    DoEvents
    Me.cmdClose.Enabled = True
End Sub
```

```
***********************************************************************
frmDocument.frm
***********************************************************************
'Copyrights - Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State
University


    Dim oBook As Excel.Workbook
    Dim oSheet As Excel.Worksheet
    Dim oSheet2 As Excel.Worksheet
    Dim oChart As Excel.Chart
    Dim oChart2 As Excel.Chart
    Dim NoG As Long
    Dim r2 As Long
    Dim c2%
    Dim TData(1 To 200, 1 To 2) As Single
    Private i As Double
    Private VV As Double
    Private II As Double

Private Sub chkGradiCheck_Click()
    If chkGradiCheck.value = 0 Then
        chkSaveTransient.Enabled = False
    Else
        chkSaveTransient.Enabled = True
    End If
End Sub

Private Sub chkSweep_Click()
    If chkSweep.value = 1 Then
        txtNoCycle.Enabled = True
        txtNoTranCycle.Enabled = True
        Label22.Enabled = True
    Else
        txtNoCycle.Enabled = False
        txtNoTranCycle.Enabled = False
        Label22.Enabled = False
    End If
End Sub

Private Sub cmbDescription_Click()
    txtCellNo = cmbDescription.ListIndex + 1
    DescripChange
End Sub

Private Sub cmbDescription_DropDown()
    DescripChange
End Sub

Private Sub cmbDescription_KeyPress(KeyAscii As Integer)
```

```
    If KeyAscii = 13 Then
        DescripChange
    End If
End Sub

Private Sub cmbDescription_LostFocus()
    DescripChange
End Sub
Sub DescripChange()
    cmbDescription.List(txtCellNo - 1) = cmbDescription.Text
    cmbDescription.ListIndex = txtCellNo - 1

    With oSheet
    If NoG > 0 Then
       ' If txtCellNo < cmbDescription.ListCount Then
       '     oChart.SeriesCollection(Val(txtCellNo.Text)).ChartType =
                                                        xlXYScatterSmooth
       ' End If
       '.Cells(3, txtCellNo * 2).value = cmbDescription.List(txtCellNo-1)
    End If
    End With
End Sub

Private Sub cmdBrowsF_Click()
    On Error Resume Next
    With CommonDialog
        .DialogTitle = "Enter File name"
        .CancelError = True
        .Filter = "Excel File (*.xls)|*.xls"
        .FileName = txtFileName
        .Flags = cdlOFNOverwritePrompt
        .ShowSave
        txtFileName = .FileName
        Caption = .FileName
        Save
    End With
End Sub

Private Sub cmdGColor_Click(Index As Integer)

    CommonDialog.ShowColor
    oBook.Colors(Index + 25) = CommonDialog.Color
    cmdGColor(Index).BackColor = CommonDialog.Color
    Exit Sub
    With oChart.SeriesCollection(Index + 1)
        .Border.Color = CommonDialog.Color
        .MarkerBackgroundColor = CommonDialog.Color
        .MarkerForegroundColor = 0
        .MarkerSize = 4
    End With

End Sub

Private Sub cmdSave_Click()
    Save
End Sub
```

```
Private Sub cmdSMCont_Click()
    frmSMCont.Visible = True
End Sub

Private Sub Form_Activate()
    If Me.optMidGND = True Then
        fMainForm.chkMidGND = 1
    Else
        fMainForm.chkMidGND = 0
    End If
    fMainForm.opt4Wire = Me.opt4Wire
    fMainForm.opt2Wire = Me.opt2Wire
End Sub

Private Sub Form_Initialize()
    Me.optMidGND.value = fMainForm.chkMidGND.value
End Sub

Private Sub Form_Load()
    Dim Para(1 To 12, 1 To 1) As Single
   ' OLE1.CreateEmbed "", "excel.chart"
    Set oBook = OLE1.object
    Set oChart = oBook.Charts(1)
    Set oChart2 = oBook.Charts(2)
    Set oSheet = oBook.Worksheets(1)
    Set oSheet2 = oBook.Worksheets(2)
    c2% = 1
    r2 = 1
    cmbDescription.List(0) = "Cell No 1"
    cmbDescription.ListIndex = 0

    For i = 0 To 19
        cmdGColor(i).BackColor = oBook.Colors(i + 25)
    Next i

    If IVTMode = False Then
        txtStartT.Enabled = False
        txtEndT.Enabled = False
        txtStepT.Enabled = False
    Else
        txtStartT.Enabled = True
        txtEndT.Enabled = True
        txtStepT.Enabled = True
    End If
    opt4Wire.value = fMainForm.opt4Wire.value
    Me.opt4Wire.value = fMainForm.opt4Wire.value
    Me.opt2Wire.value = fMainForm.opt2Wire.value

    SetSetupPara 'this subprogram assign all parameter values from frmSetup
    DoEvents

    Para(1, 1) = txtSize
    Para(2, 1) = txtStartV
    Para(3, 1) = txtEndV
    Para(4, 1) = txtStepV
    Para(5, 1) = txtCurrentlimit
```

```
    Para(6, 1) = txtDelay
    Para(7, 1) = txtAvg
    Para(8, 1) = txtNoCycle
    Para(9, 1) = 0
    Para(10, 1) = txtStartT
    Para(11, 1) = txtEndT
    Para(12, 1) = txtStepT

    With oSheet
        .Cells(2, 3).value = txtFileName
        .Cells(3, 3).value = txtDescription
        .Cells(5, 3).value = Date
        .Cells(5, 4).value = Time
        .Range("C6", "C17").value = Para

    End With

    OLE1.Visible = True
  ' OLE1.Close  'Deactivate the OLE container
  ' Application.Assistant.Visible = True
End Sub
Private Sub Form_Resize()
    On Error Resume Next
    OLE1.Move 115, 320, Me.ScaleWidth - 200, Me.ScaleHeight - 500
    freSet.Move 115, 315, Me.ScaleWidth - 200, Me.ScaleHeight - 500
    tbsGtoS.Width = Me.ScaleWidth - 200
End Sub
Sub FopenForPhotoIV(FileName As String)
    OLE1.CreateEmbed FileName, "excel.chart"
    Set oBook = OLE1.object
    Set oSheet = oBook.Worksheets(1)

    With oSheet
        If Left(.Cells(1, 1).value, 39) <> "Viraj2005 Automated IVT
                                                Setup Data File" Then
            MsgBox "This is not a correct Viraj2005 IVT Curve file",
                         vbOKOnly + vbInformation, "Invalied file format"
            Exit Sub
        End If
        Set oChart = oBook.Charts(1)
        CNo% = .Cells(4, 4).value
        lblChannelNo = CNo%
        frmSetup.TgB(CNo% - 1).value = 1
        'frmSetup.TgB_Click CNo% - 1

    End With
    For z% = 1 To NoG
        cmbDescription.List(z% - 1) = oSheet.Cells(3, 2 * z%).value
    Next z%
    cmbDescription.AddItem "Cell No " & LTrim(Str(NoG + 1))

End Sub

Sub Save()
    oSheet.Cells(4, 4).value = Val(lblChannelNo.Caption)
    If txtFileName = "" Then
        On Error GoTo ErrHandler
```

```
        With CommonDialog
            .CancelError = True
            .Filter = "Excel File (*.xls)|*.xls"
            .DialogTitle = "Save IV Curve File"
            .Flags = cdlOFNOverwritePrompt
            .ShowSave
            If Len(.FileName) = 0 Then
                Exit Sub
            End If
            txtFileName = .FileName
        End With
        Caption = txtFileName
    End If
        On Error GoTo ErrHandler
        oBook.Windows.Item(1).Visible = True
        oBook.Windows.Item(1).WindowState = xlMaximized
        oBook.SaveCopyAs txtFileName

ErrHandler:
    Exit Sub
End Sub

Sub SaveAs()
    On Error GoTo ErrHandler
    With CommonDialog
        .CancelError = True
        .Filter = "Excel File (*.xls)|*.xls"
        .DialogTitle = "Save IV Curve File As"
        .Flags = cdlOFNOverwritePrompt
        .ShowSave
        If Len(.FileName) = 0 Then
            Exit Sub
        End If
        txtFileName = .FileName
    End With
    oBook.SaveCopyAs txtFileName
    Caption = txtFileName

ErrHandler:
    Exit Sub
End Sub

Sub GetIV(Temp As Single)
Dim i As Single
Dim SCIndex As Integer 'Series collection Index
SCIndex = 1
test:
    SendSM "*RST", stat%
    If stat% <> 0 Then
       cho  =  MsgBox("KEITHLEY  2400  SourceMeter  not  found",  vbRetryCancel  +
vbExclamation, "GPIB Address Error")
       If cho = 4 Then
            GoTo test
       Else
           Exit Sub
       End If
    End If
```

```
    With oSheet
        NoG = .Cells(4, 3).value + 1
        cn% = (NoG * 5) - 4
        Rn% = 20
        If NoG > 1 Then
            Set oRng1 = oSheet.Range("A7:E20")
            Set oRng2 = oSheet.Range(.Cells(7, cn%), .Cells(20, cn% + 4))
            Call oRng1.Copy(Destination:=oRng2)
        End If
        If Temp > 0 Then
            If optDark = True Then
                .Cells(18, cn% + 1).value = Temp
            Else
                .Cells(18, cn% + 2).value = Temp
            End If
        Else
            .Cells(18, cn% + 1).value = " "
            .Cells(18, cn% + 2).value = " "
        End If

        If NoG > 1 Then
            oChart.SeriesCollection.NewSeries
            SCIndex = oChart.SeriesCollection.Count
            oChart.SeriesCollection(SCIndex).Name = .Range(.Cells(18, cn% + 1),
.Cells(19, cn% + 1))
        End If
    End With
    With oChart
        .Axes(xlCategory).MinimumScale = txtStartV.Text
        .Axes(xlCategory).MaximumScale = txtEndV.Text
    End With
    iprot$ = ":SENS:CURR:PROT " & Str(txtCurrentlimit.Text / 1000)
    VLevel$ = ":SOUR:VOLT:LEV " & txtStartV.Text
    SDelay$ = ":SOUR:DEL " & txtDelay.Text
'    Avg$ = ":SENS:AVER:COUN " & TxtAvg.Text
    SendSM ":SOUR:FUNC VOLT", stat%
    SendSM ":SOUR:VOLT:MODE FIXED", stat%
    SendSM ":SENS:FUNC 'CURR'", stat%
    SendSM iprot$, stat%
    SendSM VLevel$, stat%
    SendSM ":SENS:CURR:RANG:AUTO ON", stat%
    SendSM SDelay$, stat%
'   SendSM Avg$, stat%
'   SendSM ":SENS:AVER:STAT ON", stat%
    If opt4Wire.value = True Then
        SendSM ":SYST:RSEN ON", stat%
    Else
        SendSM ":SYST:RSEN OFF", stat%
    End If
    SendSM ":OUTPUT ON", stat%

    StartV = Val(txtStartV)
    EndV = Val(txtEndV)
    StepV = Val(txtStepV)
    NoCycle% = CInt(Val(txtNoCycle))
```

```
    c2% = 1

    For NoSweep% = 1 To NoCycle%
        r2 = 1
        If chkSweep.value = 1 Then
            oSheet2.Cells(r2, c2%).value = "time (s)"
            oSheet2.Cells(r2, c2% + 1).value = "Current(A)"
            oSheet2.Cells(r2, c2% + 2).value = "Bias Voltage (V)"
        End If
        r2 = r2 + 1

        For i = StartV To EndV Step StepV
            t = Timer()
        '   VV = 0
            ok = 0
            SendSM ":SOUR:VOLT:LEV " & Str(i), stat%
            oSheet2.Cells(r2, c2% + 2).value = Str(i)

'********************* If Gradient Check ON **************************
            If chkGradiCheck.value = 1 Then

                While ok = 0
                    k = 1
                    II = 0
                    IOld = 0
                    n = 0

                    For j = 1 To txtAvg.Text
'************* Check START/PAUSE/STOP Status*************
                        With fMainForm
                            If .tbToolBar.Buttons.Item(12).value =
                                                        tbrUnpressed Then
                                If .tbToolBar.Buttons.Item(13).value =
                                                          tbrPressed Then
                                   ' .MousePointer = 0
                                    While
.tbToolBar.Buttons.Item(13).
                                                       value = tbrPressed
                                        DoEvents
                                    Wend
                                End If
                                If .tbToolBar.Buttons.Item(12).value =
                                                        tbrUnpressed Then
                                   ' .MousePointer = 0
                                    GoTo Finalize
                                    'Exit Sub
                                End If
                            End If
                            '.MousePointer = 11
                        End With

'**********************************************************

                        SendSM "read?", stat%
                        ReadSM r$
                        TData(k, 1) = Timer() - t
```

```
                INew = Val(Mid(r$, 15, 13))
                TData(k, 2) = INew

                If IOld < INew Then n = n + 1
                If IOld > INew Then n = n - 1
                II = II + INew
                IOld = INew
                DoEvents
                k = k + 1
            Next j

            If Val(txtNoTranCycle.Text) >= NoSweep% And
                                  chkSaveTransient.value = 1 Then
                k = k - 1
                oSheet2.Range(oSheet2.Cells(r2, c2%),
                 oSheet2.Cells(r2 + k - 1, c2% + 1)).value = TData
                r2 = r2 + k
                oChart2.SeriesCollection(1).XValues =
                            oSheet2.Range(oSheet2.Cells(2, c2%),
                                       oSheet2.Cells(r2 + 1, c2%))
                oChart2.SeriesCollection(1).Values =
                         oSheet2.Range(oSheet2.Cells(2, c2% + 1),
                                  oSheet2.Cells(r2 + 1, c2% + 1))
                DoEvents
            End If
            If Abs(n) < 4 Then ok = 1
        Wend
    Else

'********************* If Gradient Check OFF ************************
        II = 0
        For j = 1 To Val(txtAvg.Text)
'************* Check START/PAUSE/STOP Status*************
            With fMainForm
                If .tbToolBar.Buttons.Item(12).value =
                                              tbrUnpressed Then
                    If .tbToolBar.Buttons.Item(13).value =
                                                tbrPressed Then
                       ' .MousePointer = 0
                        While .tbToolBar.Buttons.Item(13).value =
                                                     tbrPressed
                            DoEvents
                        Wend
                    End If
                    If .tbToolBar.Buttons.Item(12).value =
                                              tbrUnpressed Then
                       ' .MousePointer = 0
                        GoTo Finalize
                        'Exit Sub
                    End If
                End If
                '.MousePointer = 11
            End With
            '**********************************************************
            SendSM "read?", stat%
            ReadSM r$
            INew = Val(Mid(r$, 15, 13))
```

```
                    II = II + INew
                    DoEvents
                Next j
            End If
'*****************************************************************

            VV = Val(Mid(r$, 1, 13))
            II = II / txtAvg.Text
            With oSheet
                .Cells(Rn%, cn%).value = VV
                If optDark.value = True Then
                    .Cells(Rn%, cn% + 1).value = Abs(II)
                    If chkSaveIwithSign.value = 1 Then
                        .Cells(Rn%, cn% + 2).value = II
                    End If
                    DoEvents
                    oChart.SeriesCollection(SCIndex).XValues=
                                .Range(.Cells(20, cn%), .Cells(Rn%, cn%))
                    oChart.SeriesCollection(SCIndex).Values =
                        .Range(.Cells(20, cn% + 1), .Cells(Rn%, cn% + 1))
                Else
                    'oSheet.Cells(Rn%, cn% + 1).value = Abs(II)
                    oSheet.Cells(Rn%, cn% + 4).value = Abs(II)
                    DoEvents
                    oChart.SeriesCollection(SCIndex).XValues =
                                .Range(.Cells(20, cn%), .Cells(Rn%, cn%))
                    oChart.SeriesCollection(SCIndex).Values =
                        .Range(.Cells(20, cn% + 2), .Cells(Rn%, cn% + 4))
                End If
            End With
            DoEvents
            Rn% = Rn% + 1
        Next i

        TempStartV = StartV
        StartV = EndV
        EndV = TempStartV
        StepV = -StepV
        If chkSweep.value = 1 And chkSaveTransient.value = 1 Then
            c2% = c2% + 3
        End If
    Next NoSweep%

Finalize:
    SendSM ":OUTPUT OFF", stat%
    cmbDescription.List(SCIndex) = "Cell No " & LTrim(Str(NoG + 1))
    cmbDescription.ListIndex = SCIndex - 1
    oSheet.Cells(4, 3).value = NoG
    oSheet.Cells(14, cn% + 2).value = Rn% - 20
    If chkFindR.value = 1 Then
        Set oRng1 = oSheet.Range(oSheet.Cells(20, cn% + 3), oSheet.Cells(20,
cn% + 3))
        Set oRng2 = oSheet.Range(oSheet.Cells(20, cn% + 3), oSheet.Cells(Rn% -
2, cn% + 3))
        Call oRng1.AutoFill(Destination:=oRng2)
    End If
 End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
If txtFileName = "" Then
    cho = MsgBox("Do you want to save the changes on " & Me.Caption,
vbYesNoCancel + vbQuestion, "Viraj2005IVT")
    If cho = 2 Then
        Cancel = 1
        Exit Sub
    ElseIf cho = 6 Then
        Save
    End If
Else
    Save
End If
End Sub



Private Sub freTS_Click()
    If IVTMode = False Then
        txtStartT.Enabled = False
        txtEndT.Enabled = False
        txtStepT.Enabled = False
    Else
        txtStartT.Enabled = True
        txtEndT.Enabled = True
        txtStepT.Enabled = True
    End If

End Sub

Private Sub Frame5_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
    freModeFig.Visible = False
End Sub


Private Sub opt2Wire_Click()
    If opt2Wire.value = True Then
        SendSM ":SYST:RSEN OFF", stat%
        fMainForm.opt2Wire.value = True
    End If
End Sub

Private Sub opt4Wire_Click()
    If opt4Wire.value = True Then
        SendSM ":SYST:RSEN ON", stat%
        fMainForm.opt4Wire.value = True
    End If
End Sub

Private Sub optMidGND_Click()
    If optMidGND.value = True Then
        SendSW ":clos (@ 1!22)", stat%
        SendSW ":open (@ 1!21)", stat%
        fMainForm.chkMidGND.value = 1
    End If
```

```
End Sub

Private Sub optMidGND_MouseMove(Button As Integer, Shift As Integer, X As
Single, Y As Single)
    freModeFig.Visible = True
End Sub

Private Sub optSepGND_Click()
    If optMidGND.value = False Then
        SendSW ":clos (@ 1!21)", stat%
        SendSW ":open (@ 1!22)", stat%
        fMainForm.chkMidGND.value = 0
    End If

End Sub

Private Sub optSepGND_MouseMove(Button As Integer, Shift As Integer, X As
Single, Y As Single)
    freModeFig.Visible = True
End Sub


Private Sub tbsGtoS_Click()
    Select Case tbsGtoS.SelectedItem.Index
        Case 1
            freSet.Visible = False
            oChart.Activate
            OLE1.Visible = True
            oChart.Axes(xlValue).ScaleType = xlLogarithmic
        Case 2
            freSet.Visible = False
            oChart.Activate
            OLE1.Visible = True
            oChart.Axes(xlValue).ScaleType = xlLinear
        Case 3
            freSet.Visible = False
            oChart2.Activate
            OLE1.Visible = True
            oChart2.Axes(xlValue).ScaleType = xlLinear
        Case 4
            freSet.Visible = True
            OLE1.Visible = False
    End Select
End Sub

Sub SetSetupPara()
    txtStartV = frmSetup.txtStartV
    txtStepV = frmSetup.txtStepV
    txtEndV = frmSetup.txtEndV
    txtNoCycle = frmSetup.txtNoCycle
    chkSweep = frmSetup.chkSweep
    tglAutoVLimits = frmSetup.tglAutoVLimits
    txtCurrentlimit = frmSetup.txtCurrentlimit
    txtDelay = frmSetup.txtDelay
    txtAvg = frmSetup.txtAvg
    txtStartT = frmSetup.txtStartT
```

```
    txtStepT = frmSetup.txtStepT
    txtEndT = frmSetup.txtEndT
    optDark = frmSetup.optDark
    optPhoto = frmSetup.optPhoto
    optMidGND = frmSetup.optMidGND
    optSepGND = frmSetup.optSepGND
End Sub

Private Sub txtAvg_Change()
   oSheet.Cells(12, 3) = Val(txtAvg)
End Sub

Private Sub txtCurrentlimit_Change()
    If txtCurrentlimit <> "" Then
        If txtCurrentlimit > 10 Then
            cho = MsgBox("! WARNING ! Your Current Limit is grater than 10mA" &
Chr(10) & cr & "Are you sure", vbExclamation + vbYesNo, "WARNING")
            If cho <> 6 Then
                txtCurrentlimit = 10
            End If
        End If
        oSheet.Cells(10, 3) = Val(txtCurrentlimit)
    End If
End Sub

Private Sub txtDelay_Change()
   oSheet.Cells(11, 3) = Val(txtDelay)
End Sub

Private Sub txtDescription_Change()
    oSheet.Cells(3, 3) = Val(txtDescription)
End Sub

Private Sub txtEndT_Change()
    oSheet.Cells(16, 3) = Val(txtEndT)
End Sub

Private Sub txtEndV_Change()
   oSheet.Cells(8, 3) = Val(txtEndV)
End Sub

Private Sub txtFilename_Change()
    oSheet.Cells(2, 3) = Val(txtFileName)
End Sub

Private Sub txtNoCycle_Change()
    oSheet.Cells(13, 3) = Val(txtNoCycle)
End Sub

Private Sub txtSize_Change()
    oSheet.Cells(6, 3) = Val(txtSize)
End Sub

Private Sub txtStartT_Change()
   oSheet.Cells(17, 3) = Val(txtStepT)
End Sub
```

```
Private Sub txtStartV_Change()
    oSheet.Cells(7, 3) = Val(txtStartV)
End Sub

Private Sub txtStepT_Change()
    oSheet.Cells(8, 3) = Val(txtStepT)
End Sub

Private Sub txtStepV_Change()
    oSheet.Cells(9, 3) = Val(txtStepV)
End Sub

Sub FindVlimit()
    oSV = frmSetup.txtStartV
    oEV = frmSetup.txtEndV
    SendSM ":OUTPUT ON", stat%

    SendSM ":SOUR:VOLT:LEV " & Str(i), stat%
    Wait 0.2
    SendSM "read?", stat%
    ReadSM r$
'       VNew = Val(Mid(r$, 1, 13))
    INew = Val(Mid(r$, 15, 13))

End Sub


**********************************************************************
frmMain.frm
**********************************************************************
'Copyrights - Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State
University

   Begin VB.Menu mnuFile
      Caption         =   "&File"
      Begin VB.Menu mnuFileNew
         Caption         =   "&New"
         Shortcut        =   ^N
      End
      Begin VB.Menu mnuFileOpen
         Caption         =   "&Open..."
         Shortcut        =   ^O
      End
      Begin VB.Menu mnuFileClose
         Caption         =   "&Close"
      End
      Begin VB.Menu mnuFileBar0
         Caption         =   "-"
      End
      Begin VB.Menu mnuFileSave
         Caption         =   "&Save"
      End
      Begin VB.Menu mnuFileSaveAs
         Caption         =   "Save &As..."
      End
      Begin VB.Menu mnuFileSaveAll
         Caption         =   "Save A&ll"
```

```
      End
      Begin VB.Menu mnuFileBar1
         Caption         =   "-"
      End
      Begin VB.Menu mnuFileProperties
         Caption         =   "Propert&ies"
      End
      Begin VB.Menu mnuFileBar2
         Caption         =   "-"
      End
      Begin VB.Menu mnuFilePageSetup
         Caption         =   "Page Set&up..."
      End
      Begin VB.Menu mnuFilePrintPreview
         Caption         =   "Print Pre&view"
      End
      Begin VB.Menu mnuFilePrint
         Caption         =   "&Print..."
      End
      Begin VB.Menu mnuFileBar3
         Caption         =   "-"
      End
      Begin VB.Menu mnuFileSend
         Caption         =   "Sen&d..."
      End
      Begin VB.Menu mnuFileBar4
         Caption         =   "-"
      End
      Begin VB.Menu mnuFileMRU
         Caption         =   ""
         Index           =   1
         Visible         =   0   'False
      End
      Begin VB.Menu mnuFileMRU
         Caption         =   ""
         Index           =   2
         Visible         =   0   'False
      End
      Begin VB.Menu mnuFileMRU
         Caption         =   ""
         Index           =   3
         Visible         =   0   'False
      End
      Begin VB.Menu mnuFileBar5
         Caption         =   "-"
         Visible         =   0   'False
      End
      Begin VB.Menu mnuFileExit
         Caption         =   "E&xit"
      End
   End
   Begin VB.Menu mnuEdit
      Caption         =   "&Edit"
      Begin VB.Menu mnuEditUndo
         Caption         =   "&Undo"
      End
      Begin VB.Menu mnuEditBar0
```

```
      Caption         =   "-"
   End
   Begin VB.Menu mnuEditCut
      Caption         =   "Cu&t"
      Shortcut        =   ^X
   End
   Begin VB.Menu mnuEditCopy
      Caption         =   "&Copy"
      Shortcut        =   ^C
   End
   Begin VB.Menu mnuEditPaste
      Caption         =   "&Paste"
      Shortcut        =   ^V
   End
   Begin VB.Menu mnuEditPasteSpecial
      Caption         =   "Paste &Special..."
   End
End
Begin VB.Menu mnuView
   Caption         =   "&View"
   Begin VB.Menu mnuViewToolbar
      Caption         =   "&Toolbar"
      Checked         =   -1  'True
   End
   Begin VB.Menu mnuViewStatusBar
      Caption         =   "Status &Bar"
      Checked         =   -1  'True
   End
   Begin VB.Menu mnuViewBar0
      Caption         =   "-"
   End
   Begin VB.Menu mnuViewRefresh
      Caption         =   "&Refresh"
   End
   Begin VB.Menu mnuViewOptions
      Caption         =   "&Options..."
   End
   Begin VB.Menu mnubar2
      Caption         =   "-"
   End
   Begin VB.Menu mnuSMController
      Caption         =   "Source &Meter Controller"
   End
   Begin VB.Menu mnuSWsystem
      Caption         =   "&Switch System"
   End
   Begin VB.Menu mnuTempController
      Caption         =   "&Temperature Controller"
   End
End
Begin VB.Menu mnuTools
   Caption         =   "&Tools"
   Begin VB.Menu mnuInitializeSM
      Caption         =   "Initialize Source &Meter"
   End
   Begin VB.Menu mnuInitializeSW
      Caption         =   "Initialize &Switch System"
```

```
      End
      Begin VB.Menu mnuInitializeTC
         Caption         =   "Initialize &Temp. Controller"
      End
      Begin VB.Menu mnubar3
         Caption         =   "-"
      End
      Begin VB.Menu mnuSetupScane
         Caption         =   "Setup Scan"
         Begin VB.Menu mnuIVT
            Caption         =   "IVT"
         End
         Begin VB.Menu mnuIV
            Caption         =   "IV"
         End
      End
      Begin VB.Menu mnubar6
         Caption         =   "-"
      End
      Begin VB.Menu mnuGPIB
         Caption         =   "Change GPIB Address"
         Begin VB.Menu mnuGPIBSM
            Caption         =   "KEITHLEY 2400 Source Meter"
         End
         Begin VB.Menu mnuGPIBSW
            Caption         =   "KEITHLEY 7001 Switch System"
         End
         Begin VB.Menu mnuGPIBTC
            Caption         =   "SI 9620-1 Temp. Controller"
         End
      End
   End
   Begin VB.Menu mnuWindow
      Caption         =   "&Window"
      WindowList      =   -1  'True
      Begin VB.Menu mnuWindowNewWindow
         Caption         =   "&New Window"
      End
      Begin VB.Menu mnuWindowBar0
         Caption         =   "-"
      End
      Begin VB.Menu mnuWindowCascade
         Caption         =   "&Cascade"
      End
      Begin VB.Menu mnuWindowTileHorizontal
         Caption         =   "Tile &Horizontal"
      End
      Begin VB.Menu mnuWindowTileVertical
         Caption         =   "Tile &Vertical"
      End
      Begin VB.Menu mnuWindowArrangeIcons
         Caption         =   "&Arrange Icons"
      End
   End
   Begin VB.Menu mnuHelp
      Caption         =   "&Help"
      Begin VB.Menu mnuHelpContents
```

```
         Caption         =   "&Contents"
      End
      Begin VB.Menu mnuHelpSearchForHelpOn
         Caption         =   "&Search For Help On..."
      End
      Begin VB.Menu mnuHelpBar0
         Caption         =   "-"
      End
      Begin VB.Menu mnuHelpAbout
         Caption         =   "&About "
      End
   End
End
Attribute VB_Name = "frmMain"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Private Declare Function SendMessage Lib "user32" Alias "SendMessageA" (ByVal
hwnd As Long, ByVal wMsg As Long, ByVal wParam As Long, ByVal lParam As Any) As
Long
Const EM_UNDO = &HC7
Private Declare Function OSWinHelp% Lib "user32" Alias "WinHelpA" (ByVal hwnd&,
ByVal HelpFile$, ByVal wCommand%, dwData As Any)
Dim Form(1 To 10) As frmDocument
Dim TvstForm As frmTempVsTime
Dim TvstDoc As frmTempVsTime
Dim DFileName(1 To 10) As String
Private cr As String
Private NoChanal As Integer
Dim Pause As Boolean

Private Sub chkMidGND_Click()
    On Error GoTo errorhandler
    If chkMidGND.value = 1 Then
        SendSW ":clos (@ 1!22)", stat%
        SendSW ":open (@ 1!21)", stat%
        Me.ActiveForm.optMidGND = 1
    Else
        SendSW ":clos (@ 1!21)", stat%
        SendSW ":open (@ 1!22)", stat%
        Me.ActiveForm.optSepGND = 1
    End If

errorhandler:
End Sub

Private Sub cmdSMControl_Click()
    frmSMCont.Show vbModal, Me
End Sub

Private Sub cmdSwitch_MouseDown(Button As Integer, Shift As Integer, X As
Single, Y As Single)
    If cmdSwitch.Top = 180 Then
        cmdSwitch.Top = 720
        IVTMode = False
    Else
```

```
        cmdSwitch.Top = 180
        IVTMode = True
    End If

End Sub


Private Sub MDIForm_Load()
    Me.Left = GetSetting(App.Title, "Settings", "MainLeft", 1000)
    Me.Top = GetSetting(App.Title, "Settings", "MainTop", 1000)
    Me.Width = GetSetting(App.Title, "Settings", "MainWidth", 6500)
    Me.Height = GetSetting(App.Title, "Settings", "MainHeight", 6500)
    LoadNewDoc
    cr = Chr(13)
    IVTMode = True
    NoChanal = 0

End Sub

Private Sub LoadNewDoc()
    Static lDocumentCount As Long
    Dim frmD As frmDocument
    lDocumentCount = lDocumentCount + 1
    Set frmD = New frmDocument
    frmD.Caption = "IV Curve Book " & lDocumentCount
    frmD.Width = 9480
    frmD.Height = Me.Height - 2700
    frmD.Show
End Sub

Private Sub MDIForm_Unload(Cancel As Integer)
    Close All
    If Me.WindowState <> vbMinimized Then
        SaveSetting App.Title, "Settings", "MainLeft", Me.Left
        SaveSetting App.Title, "Settings", "MainTop", Me.Top
        SaveSetting App.Title, "Settings", "MainWidth", Me.Width
        SaveSetting App.Title, "Settings", "MainHeight", Me.Height
    End If
    End
End Sub

Private Sub mnuGPIBSM_Click()
    X = InputBox("Enter New GPIB Address of KEITHLEY 2400 Source Meter",
"Change GPIB Address", GPIBSM)
    If X > 0 And X < 25 Then
        GPIBSM = X
        SaveSetting App.Title, "Settings", "GPIBSM", GPIBSM
    End If
End Sub

Private Sub mnuGPIBSW_Click()
    X = InputBox("Enter New GPIB Address of KEITHLEY 7001 Switch System",
"Change GPIB Address", GPIBSW)
    If X > 0 And X < 25 Then
        GPIBSW = X
        SaveSetting App.Title, "Settings", "GPIBSW", GPIBSW
    End If
```

```
End Sub

Private Sub mnuGPIBTC_Click()
    X = InputBox("Enter New GPIB Address of SI 9620-1 Temp. Controller",
"Change GPIB Address", GPIBTC)
    If X > 0 And X < 30 Then
        GPIBTC = X
        SaveSetting App.Title, "Settings", "GPIBTC", GPIBTC
    End If
End Sub

Private Sub mnuSMController_Click()
    frmSMCont.Visible = True
End Sub

Private Sub mnuSWsystem_Click()
    Keithley7001.Visible = True
End Sub

Private Sub mnuTempController_Click()
    TempController.Visible = True
End Sub

Private Sub opt2Wire_Click()
    On Error GoTo errorhandler
    If opt2Wire.value = True Then
        SendSM ":SYST:RSEN OFF", stat%
        Me.ActiveForm.opt2Wire = 1
    End If
errorhandler:
End Sub

Private Sub opt4Wire_Click()
    On Error GoTo errorhandler
    If opt4Wire.value = True Then
        SendSM ":SYST:RSEN ON", stat%
        Me.ActiveForm.opt4Wire = 1
    End If
errorhandler:
End Sub

Private Sub tbToolBar_ButtonClick(ByVal Button As MSComctlLib.Button)
    On Error Resume Next
    Select Case Button.Key
        Case "New"
            LoadNewDoc

        Case "Open"
            mnuFileOpen_Click

        Case "Save"
            mnuFileSave_Click

        Case "Print"
            If Form(1).Visible = False Then
                MsgBox "aa"
            End If
```

```
        Case "Initialize"
            frmMessage.Show

        Case "Setup"
            frmSetup.Show vbModal, Me

        Case "Options"

        Case "Settings"
            frmOptions.Show vbModal, Me

        Case "Start"
            If Pause = False Then
                RunScan
            Else
                Pause = False
            End If
        Case "Pause"
            If tbToolBar.Buttons.Item(13).value = tbrPressed Then
                Pause = True
            Else
                Pause = False
            End If
        Case "Stop"
            Button.value = tbrUnpressed
            TvstForm.cmdClose.Enabled = True
            Toolbar1.Enabled = True
            fMainForm.Frame2.Enabled = True
        Case "TempCont"
            TempController.Visible = True

        Case "ShowT"
            If Button.value = tbrPressed Then
                txtlblT1.Visible = True
                txtT1.Visible = True
                txtlblT2.Visible = True
                txtT2.Visible = True
                txtlblH.Visible = True
                txth.Visible = True
                Set TvstDoc = New frmTempVsTime
                TvstDoc.Height = 6330
                TvstDoc.Width = 8175
                TvstDoc.Show
                tmrTemp.Enabled = True
            Else
                txtlblT1.Visible = False
                txtT1.Visible = False
                txtlblT2.Visible = False
                txtT2.Visible = False
                txtlblH.Visible = False
                txth.Visible = False
                tmrTemp.Enabled = False

            End If
    End Select
End Sub
```

```
Private Sub mnuHelpAbout_Click()
    frmAbout.Show vbModal, Me
End Sub

Private Sub mnuHelpSearchForHelpOn_Click()
    Dim nRet As Integer


    'if there is no helpfile for this project display a message to the user
    'you can set the HelpFile for your application in the
    'Project Properties dialog
    If Len(App.HelpFile) = 0 Then
        MsgBox "Unable to display Help Contents. There is no Help associated
with this project.", vbInformation, Me.Caption
    Else
        On Error Resume Next
        nRet = OSWinHelp(Me.hwnd, App.HelpFile, 261, 0)
        If Err Then
            MsgBox Err.Description
        End If
    End If

End Sub

Private Sub mnuHelpContents_Click()
    Dim nRet As Integer


    'if there is no helpfile for this project display a message to the user
    'you can set the HelpFile for your application in the
    'Project Properties dialog
    If Len(App.HelpFile) = 0 Then
        MsgBox "Unable to display Help Contents. There is no Help associated
with this project.", vbInformation, Me.Caption
    Else
        On Error Resume Next
        nRet = OSWinHelp(Me.hwnd, App.HelpFile, 3, 0)
        If Err Then
            MsgBox Err.Description
        End If
    End If

End Sub


Private Sub mnuWindowArrangeIcons_Click()
    Me.Arrange vbArrangeIcons
End Sub

Private Sub mnuWindowTileVertical_Click()
    Me.Arrange vbTileVertical
End Sub

Private Sub mnuWindowTileHorizontal_Click()
    Me.Arrange vbTileHorizontal
End Sub
```

```
Private Sub mnuWindowCascade_Click()
    Me.Arrange vbCascade
End Sub

Private Sub mnuWindowNewWindow_Click()
    LoadNewDoc
End Sub

Private Sub mnuViewOptions_Click()
    frmOptions.Show vbModal, Me
End Sub

Private Sub mnuViewRefresh_Click()
    'ToDo: Add 'mnuViewRefresh_Click' code.
    MsgBox "Add 'mnuViewRefresh_Click' code."
End Sub

Private Sub mnuViewStatusBar_Click()
    mnuViewStatusBar.Checked = Not mnuViewStatusBar.Checked
    sbStatusBar.Visible = mnuViewStatusBar.Checked
End Sub

Private Sub mnuViewToolbar_Click()
    mnuViewToolbar.Checked = Not mnuViewToolbar.Checked
    tbToolBar.Visible = mnuViewToolbar.Checked
End Sub

Private Sub mnuEditPasteSpecial_Click()
    'ToDo: Add 'mnuEditPasteSpecial_Click' code.
    MsgBox "Add 'mnuEditPasteSpecial_Click' code."
End Sub

Private Sub mnuEditPaste_Click()
    On Error Resume Next
  '  ActiveForm.rtfText.SelRTF = Clipboard.GetText

End Sub

Private Sub mnuEditCopy_Click()
    On Error Resume Next
   ' Clipboard.SetText ActiveForm.rtfText.SelRTF

End Sub

Private Sub mnuEditCut_Click()
    On Error Resume Next
   ' Clipboard.SetText ActiveForm.rtfText.SelRTF
  '  ActiveForm.rtfText.SelText = vbNullString

End Sub

Private Sub mnuEditUndo_Click()
    'ToDo: Add 'mnuEditUndo_Click' code.
    MsgBox "Add 'mnuEditUndo_Click' code."
End Sub

Private Sub mnuFileExit_Click()
```

```
    'unload the form
    Unload Me
End Sub

Private Sub mnuFileSend_Click()
    'ToDo: Add 'mnuFileSend_Click' code.
    MsgBox "Add 'mnuFileSend_Click' code."
End Sub

Private Sub mnuFilePrint_Click()
    On Error Resume Next
    If ActiveForm Is Nothing Then Exit Sub

    With dlgCommonDialog
        .DialogTitle = "Print"
        .CancelError = True
        .Flags = cdlPDReturnDC + cdlPDNoPageNums
     '   If ActiveForm.rtfText.SelLength = 0 Then
     '       .Flags = .Flags + cdlPDAllPages
     '   Else
     '       .Flags = .Flags + cdlPDSelection
     '   End If
        .ShowPrinter
'        If err <> MSComDlg.cdlCancel Then
'            ActiveForm.rtfText.SelPrint .hDC
'        End If
    End With

End Sub

Private Sub mnuFilePrintPreview_Click()
    'ToDo: Add 'mnuFilePrintPreview_Click' code.
    MsgBox "Add 'mnuFilePrintPreview_Click' code."
End Sub

Private Sub mnuFilePageSetup_Click()
    On Error Resume Next
    With dlgCommonDialog
        .DialogTitle = "Page Setup"
        .CancelError = True
        .ShowPrinter
    End With

End Sub


Private Sub mnuFileSaveAll_Click()
    'ToDo: Add 'mnuFileSaveAll_Click' code.
    MsgBox "Add 'mnuFileSaveAll_Click' code."
End Sub

Private Sub mnuFileSaveAs_Click()
    If ActiveForm.Caption <> "Viraj2003 Multimeter" Then
        ActiveForm.SaveAs
    Else
        frmMultimeter.WindowState = 1
        ActiveForm.SaveAs
```

```
        frmMultimeter.WindowState = 0
    End If
End Sub

Private Sub mnuFileSave_Click()
    If ActiveForm.Caption <> "Viraj2003 Multimeter" Then
        ActiveForm.Save
    Else
        frmMultimeter.WindowState = 1
        ActiveForm.Save
        frmMultimeter.WindowState = 0
    End If
    End Sub

Private Sub mnuFileClose_Click()
    'ToDo: Add 'mnuFileClose_Click' code.
    MsgBox "Add 'mnuFileClose_Click' code."
End Sub

Private Sub mnuFileOpen_Click()
    OpenForPhotoIV
End Sub

Private Sub mnuFileNew_Click()
    LoadNewDoc
End Sub

Public Sub multioff()
tbToolBar.Buttons.Item(5).value = tbrUnpressed
End Sub

Private Sub tglIVt_GotFocus()

End Sub

Private Sub TgB_Click(Index As Integer)
    If TgB(Index).value = True Then
        lblLED(Index).BackColor = &HFF00&
        DoEvents
        For i = 0 To 9
            If i - Index <> 0 Then TgB(i).value = False
        Next i
        CloseSW Index + 1, chkMidGND
    Else
        lblLED(Index).BackColor = &H0&
        CloseSW 0, chkMidGND
    End If

End Sub

Private Sub tmrTemp_Timer()
    txtT1.Text = Format(Mid(ReadTC("T" & cr), 2, 5), "000.0K")
    txtT2.Text = Format(Mid(ReadTC("t" & cr), 2, 5), "000.0K")
    txth.Text = Format(Mid(ReadTC("H" & cr), 2, 5), "000.0 W")
    TvstDoc.FillData Val(txtT1.Text), Val(txtT2.Text), False
End Sub
```

```
Sub OpenNewDoc(n As Integer, DFileName() As String, mesaSize() As Single)
    NoChanal = n
    For i = 1 To n
        Set Form(i) = New frmDocument
        Form(i).txtSize = mesaSize(i)
        Form(i).txtFilename = DFileName(i)
        Form(i).Caption = Mid(DFileName(i), Len(DFileName(i)) - 4, 1) & "  | "
& DFileName(i)
        Form(i).Width = 9480
        Form(i).Height = Me.Height - 2700
        Form(i).Show
        Form(i).opt4Wire.value = True
        Form(i).lblChannelNo = Asc(Mid(DFileName(i), Len(DFileName(i)) - 4, 1))
- 64
    Next i
    SaveProject
    For i = n To 2 Step -1
        Form(i).WindowState = 1
        DoEvents
    Next i

    cho = MsgBox("Do you want START Scan now?" & Chr(10) & cr & "If you select
<No> you can Start it later using Start key", vbQuestion + vbYesNo, "START it
Now?")
    If cho = 6 Then
        tbToolBar.Buttons.Item(12).value = tbrPressed 'Start button press
        RunScan
    End If

End Sub
Sub OpenForPhotoIV()
    Dim fnames As Variant
    On Error GoTo ErrHandler

    fnames = Application.GetOpenFilename("Excel Files (*.xls), *.xls", , , ,
MultiSelect:=True)
    NoChanal = UBound(fnames)

    For i = 0 To 9
        frmSetup.TgB(i).value = 0
    Next i

    Counter = 1
    While Counter <= UBound(fnames)
        Set Form(Counter) = New frmDocument
        Form(Counter).Caption = fnames(Counter)
        Form(Counter).Height = Me.Height - 2700
        Form(Counter).Width = 9480
        Form(Counter).Show
        Form(Counter).FopenForPhotoIV (fnames(Counter))

        Counter = Counter + 1
    Wend
ErrHandler:

End Sub
Sub SaveProject()
```

```
    For i = 1 To NoChanal
        Form(i).Save
    Next i
End Sub

Sub RunScan()

'############################################################################
'If Scan was setup and IVT mode is ON
'############################################################################
    Dim Temp As Single
    Frame2.Enabled = False
    If NoChanal > 0 And IVTMode = True Then
        Set TvstForm = New frmTempVsTime
        TvstForm.Height = 6330
        TvstForm.Width = 8175
        TvstForm.Top = 15
        TvstForm.Left = 6045

        TvstForm.Show
        TvstForm.Save
        txtlblT1.Visible = True
        txtT1.Visible = True
        txtlblT2.Visible = True
        txtT2.Visible = True
        txtlblH.Visible = True
        txth.Visible = True

        Toolbar1.Enabled = False
        tbToolBar.Buttons.Item(17).value = tbrPressed

        SendSM ":SYST:RSEN ON", stat%

        For    Temp    =    frmSetup.txtStartT    To    frmSetup.txtEndT    Step
frmSetup.txtStepT

            TvstForm.OLE1.MousePointer = 11
            TvstForm.SetTemp (Temp)  'To set the temp
            TvstForm.OLE1.MousePointer = 0
            For Chanel = 1 To NoChanal

                '************* Check START/PAUSE/STOP Status*************
                If tbToolBar.Buttons.Item(12).value = tbrUnpressed Then
                    If tbToolBar.Buttons.Item(13).value = tbrPressed Then
                        While tbToolBar.Buttons.Item(13).value =
                                                               tbrPressed
                            DoEvents
                        Wend
                    End If
                    If tbToolBar.Buttons.Item(12).value = tbrUnpressed
                                                                     Then
                        Exit Sub
                    End If
                End If
                '*******************************************************
                CNo% = Asc(frmSetup.TabStrip1.Tabs(Chanel).Caption) - 64
                CloseSW CNo%, frmSetup.optMidGND
```

```
                PushChannel CNo%
                Form(Chanel).WindowState = 0

                fMainForm.tmrTvst.Enabled = True
                Form(Chanel).OLE1.MousePointer = 11
                Form(Chanel).GetIV Temp
                Form(Chanel).OLE1.MousePointer = 0
                fMainForm.tmrTvst.Enabled = False

                TvstForm.FillData Val(txtT1.Text), Val(txtT2.Text), True

                Form(Chanel).Save
                Form(Chanel).WindowState = 1
            Next Chanel

        Next Temp
        TvstForm.Save
        TvstForm.cmdClose.Enabled = True
        tbToolBar.Buttons.Item(17).value = tbrUnpressed

        '************* To find Heter ON/OFF and set it to OFF
        SendTC "S3000" & cr, stat%
        SendTC "X" & cr, stat%
        Wait 1
        If Val(Mid(ReadTC("H" & cr), 2, 5)) > 0 Then
            SendTC "X" & cr, stat%
        End If
        '***************
        SendSM ":SYST:RSEN OFF", stat%
        For Chanel = 1 To NoChanal
            Form(Chanel).WindowState = 0
        Next Chanel

    End If

'#######################################################################
'If Scan was setup and IV mode is ON
'#######################################################################
    If NoChanal > 0 And IVTMode = False Then
        SendSM ":SYST:RSEN ON", stat%
        For Chanel = 1 To NoChanal
            '************* Check START/PAUSE/STOP Status*************
            If tbToolBar.Buttons.Item(12).value = tbrUnpressed Then
                If tbToolBar.Buttons.Item(13).value = tbrPressed Then
                    While tbToolBar.Buttons.Item(13).value = tbrPressed
                        DoEvents
                    Wend
                End If
                If tbToolBar.Buttons.Item(12).value = tbrUnpressed Then
                    Exit Sub
                End If
            End If
            '*********************************************************

            CNo% = Asc(frmSetup.TabStrip1.Tabs(Chanel).Caption) - 64
            CloseSW CNo%, frmSetup.optMidGND
            PushChannel CNo%
```

```
            Form(Chanel).WindowState = 0

            Form(Chanel).OLE1.MousePointer = 11
            Form(Chanel).GetIV 0
            Form(Chanel).OLE1.MousePointer = 0

            Form(Chanel).Save
            Form(Chanel).WindowState = 1
            'MsgBox CNo%
        Next Chanel
        SendSM ":SYST:RSEN OFF", stat%
        For Chanel = 1 To NoChanal
            Form(Chanel).WindowState = 0
        Next Chanel

    End If

'######################################################################
'If Scan was not setup (Normal IV curve Mode)
'######################################################################
    If NoChanal = 0 Then
        Me.ActiveForm.OLE1.MousePointer = 11
        Me.ActiveForm.GetIV 0
        Me.ActiveForm.OLE1.MousePointer = 0
    End If
    tbToolBar.Buttons.Item(12).value = tbrUnpressed
    SendSM ":OUTPUT OFF", stat%
    Frame2.Enabled = True

End Sub
Sub UpdateTemp(T1 As Single, T2 As Single, H As Single)
   txtT1.Text = Format(T1, "000.0K")
   txtT2.Text = Format(T2, "000.0K")
   txth.Text = Format(H, "000.0 W")
End Sub

Private Sub tmrTvst_Timer()
    txtT1.Text = Format(Mid(ReadTC("T" & cr), 2, 5), "000.0K")
    txtT2.Text = Format(Mid(ReadTC("t" & cr), 2, 5), "000.0K")
    txth.Text = Format(Mid(ReadTC("H" & cr), 2, 5), "000.0 W")
    TvstForm.FillData Val(txtT1.Text), Val(txtT2.Text), False
End Sub
Sub PushChannel(i As Integer)
    TgB(i - 1).value = True
    lblLED(i - 1).BackColor = &HFF00&
    For j = 0 To 9
        If j - (i - 1) <> 0 Then
            TgB(j).value = False
            lblLED(j).BackColor = &H0&
        End If
    Next j
'    CloseSW Index + 1, chkMidGND
    DoEvents
End Sub
```

```
********************************************************************frmSetup
.frm
********************************************************************
'Copyrights - Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State
University

Dim FileName(1 To 10) As String
Dim mesaSize(1 To 10) As Single

Private Sub chkSweep_Click()
    If chkSweep.value = 1 Then
        txtNoCycle.Enabled = True
    Else
        txtNoCycle.Enabled = False
    End If
End Sub

Private Sub cmdBrows_Click()
    On Error Resume Next
    With CommonDialog1
        .DialogTitle = "Enter New Project Name"
        .CancelError = True
        .Filter = "All File (*.*)|*.*"
        .Flags = cdlOFNOverwritePrompt
        .ShowSave
        txtProjectName.Text = .FileName
    End With

    If Right(txtProjectName.Text, 4) = ".xls" Then
        txtProjectName.Text             =            Left(txtProjectName.Text,
Len(txtProjectName.Text) - 6)
    End If

    For i = 1 To TabStrip1.Tabs.Count
        FileName(i) = txtProjectName.Text & "_" & TabStrip1.Tabs(i).Caption &
".xls"
    Next i
    cmdStartNewScan.Enabled = True
    TabStrip1_Click

End Sub

Private Sub cmdBrowsF_Click()
    On Error Resume Next
    With CommonDialog1
        .DialogTitle = "Measa File name"
        .CancelError = True
        .Filter = "Excel File (*.xls)|*.xls"
        .FileName = FileName(TabStrip1.SelectedItem.Index)
        .Flags = cdlOFNOverwritePrompt
        .ShowSave
        .InitDir = FileName(TabStrip1.SelectedItem.Index)
        FileName(TabStrip1.SelectedItem.Index) = .FileName
        TabStrip1_Click
    End With
End Sub
```

```
Private Sub cmdClose_Click()
    Me.Visible = False
End Sub

Private Sub cmdGetSett_Click()
    ReadFormData Me, App.Path & "\Settings.dat"
    txtProjectName = ""
End Sub

Private Sub cmdSaveSett_Click()
    SaveFormData Me, App.Path & "\Settings.dat"
End Sub

Private Sub cmdSM_Click()
    frmSMCont.Frame3.Enabled = True
    frmSMCont.Show vbModal, Me
End Sub

Private Sub cmdStartNewScan_Click()
    Me.Visible = False
    DoEvents
    If optExcelFile.value = True Then
        fMainForm.OpenNewDoc TabStrip1.Tabs.Count, FileName(), mesaSize()
    Else
        frmDataCollecting.Visible = True
        frmDataCollecting.IVFNames TabStrip1.Tabs.Count, FileName(), mesaSize()

    End If
End Sub

Private Sub Form_Activate()
    Me.opt4Wire.value = fMainForm.opt4Wire.value
    Me.opt2Wire.value = fMainForm.opt2Wire.value
    If fMainForm.chkMidGND.value = 1 Then
        Me.optMidGND.value = True
        Me.optSepGND.value = False
    Else
        Me.optMidGND.value = False
        Me.optSepGND.value = True
    End If
End Sub

Private Sub Form_Load()
    For i = 1 To 10
        mesaSize(i) = 1
    Next i
    ReadFormData Me, App.Path & "\Settings.dat"
    txtProjectName = ""
End Sub

Private Sub Form_Unload(Cancel As Integer)
    SaveFormData Me, App.Path & "\Settings.dat"
End Sub

Private Sub Frame5_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
```

```
    freModeFig.Visible = False
End Sub

Private Sub opt2Wire_Click()
    If opt2Wire.value = True Then
        SendSM ":SYST:RSEN OFF", stat%
        fMainForm.opt2Wire.value = True
    End If
End Sub

Private Sub opt4Wire_Click()
    If opt4Wire.value = True Then
        SendSM ":SYST:RSEN ON", stat%
        fMainForm.opt4Wire.value = True
    End If
End Sub

Private Sub optMidGND_Click()
    If optMidGND.value = True Then
        SendSW ":clos (@ 1!22)", stat%
        SendSW ":open (@ 1!21)", stat%
        fMainForm.chkMidGND.value = 1
    End If

End Sub

Private  Sub  optMidGND_MouseMove(Button  As  Integer,  Shift  As  Integer,  X  As
Single, Y As Single)
    freModeFig.Visible = True
End Sub

Private Sub optSepGND_Click()
    If optMidGND.value = False Then
        SendSW ":clos (@ 1!21)", stat%
        SendSW ":open (@ 1!22)", stat%
        fMainForm.chkMidGND.value = 0
    End If
End Sub

Private  Sub  optSepGND_MouseMove(Button  As  Integer,  Shift  As  Integer,  X  As
Single, Y As Single)
    freModeFig.Visible = True
End Sub

Private Sub optSize_Click(Index As Integer)
    mesaSize(TabStrip1.SelectedItem.Index)  =  (400  +  (200  *  Index))  ^  2  /
1000000
    txtSize = mesaSize(TabStrip1.SelectedItem.Index)
End Sub

Private Sub TabStrip1_Click()
    If Len(FileName(TabStrip1.SelectedItem.Index)) > 65 Then
       txtFilename.Text = "..." & Right(FileName(TabStrip1.SelectedItem.Index),
65)
    Else
       txtFilename.Text = FileName(TabStrip1.SelectedItem.Index)
    End If
```

```
    txtSize = mesaSize(TabStrip1.SelectedItem.Index)
End Sub

Sub TgB_Click(Index As Integer)
    If TgB(Index).value = True Then
        lblLED(Index).BackColor = &HFF00&
        TabStrip1.Tabs.Add , , TgB(Index).Caption
        For i = 1 To TabStrip1.Tabs.Count
            FileName(i) = txtProjectName.Text & "_" & TabStrip1.Tabs(i).Caption
& ".xls"
        Next i
    Else
        lblLED(Index).BackColor = &H0&
        For i = 1 To TabStrip1.Tabs.Count
            If TabStrip1.Tabs.Item(i).Caption = TgB(Index).Caption Then
                TabStrip1.Tabs.Remove i
                FileName(i) = ""
                Exit For
            End If
        Next i
    End If

End Sub


Private Sub txtCurrentlimit_Change()
    If txtCurrentLimit <> "" Then
        If txtCurrentLimit > 10 Then
            cho = MsgBox("! WARNING ! Your Current Limit is grater than 10mA" &
Chr(10) & cr & "Are you sure", vbExclamation + vbYesNo, "WARNING")
            If cho <> 6 Then
                txtCurrentLimit = 10
            End If
        End If
    End If
End Sub

Private Sub txtd_Change()
    txtSize = Format(4 * Atn(1) * (Val(txtd) / 2) ^ 2 / 1000000, "0.0000")
End Sub

Private Sub txth_Change()
    txtSize = Val(txth) * Val(txtw) / 1000000
End Sub

Private Sub txtProjectName_Change()
    If txtProjectName.Text <> "" Then
        cmdStartNewScan.Enabled = True
        TabStrip1_Click
    Else
        cmdStartNewScan.Enabled = False
    End If
End Sub

Private Sub txtSize_Change()
    mesaSize(TabStrip1.SelectedItem.Index) = Val(txtSize)
End Sub
```

```
Private Sub txtw_Change()
    txtSize = Val(txth) * Val(txtw) / 1000000
End Sub



**********************************************************************
frmSMCon.frm
**********************************************************************
'Copyrights - Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State University

Private Sub cmdClose_Click()
  '  SendSM ":OUTPUT OFF", stat%
    SetV = chkSetVLimits
    SV = Val(txtVStart)
    EV = Val(txtVEnd)
    Unload Me
    If frmSetup.Visible And SetV = 1 Then
        frmSetup.txtStartV = SV
        frmSetup.txtEndV = EV
    ElseIf SetV = 1 Then
        fMainForm.ActiveForm.txtStartV = SV
        fMainForm.ActiveForm.txtEndV = EV
    End If
End Sub

Private Sub cmdOutPutOnOff_Click()
    If cmdOutPutOnOff.Caption = "OutPut ON" Then
        cmdOutPutOnOff.Caption = "OutPut OFF"
        lblONOffLed.BackColor = &HFF&
        SendSM ":OUTPUT ON", stat%
        Timer1.Enabled = True
    Else
        cmdOutPutOnOff.Caption = "OutPut ON"
        lblONOffLed.BackColor = &O0&
        SendSM ":OUTPUT OFF", stat%
        lblV.Caption = Format(0, "#0.00000")
        lblI.Caption = Format(0, "###0.00000")
        Timer1.Enabled = False
    End If

End Sub


Private Sub cmdVset_Click()
    VLevel$ = ":SOUR:VOLT:LEV " & Val(txtVset)
    SendSM VLevel$, stat%
End Sub

Private Sub Form_Load()
    SetLimits = False
    SendSM ":OUTPUT OFF", stat%
    SendSM ":SOUR:FUNC VOLT", stat%
    SendSM ":SOUR:VOLT:MODE FIXED", stat%
    SendSM ":SOUR:VOLT:RANG:AUTO ON", stat%
```

```
    SendSM ":SENS:FUNC 'CURR'", stat%
    SendSM ":SENS:CURR:RANG:AUTO ON", stat%
    SendSM ":SOUR:DEL 0.1", stat%
    SendSM ":SOUR:VOLT:LEV 0.0", stat%
    iprot$ = ":SENS:CURR:PROT " & Str(txtCurrentLimit.Text / 1000)
    SendSM iprot$, stat%
End Sub



Private Sub optFindV_Click()
    If cmdOutPutOnOff.Caption = "OutPut ON" Then
        cmdOutPutOnOff_Click
    End If
    SendSM ":OUTPUT ON", stat%
    ILimit = Val(txtCurrentLimit.Text)
    iprot$ = ":SENS:CURR:PROT " & Str(ILimit / 1000)
    SendSM iprot$, stat%

    i = 0
    Do
        VLevel$ = ":SOUR:VOLT:LEV " & Str(i)
        SendSM VLevel$, stat%
        i = i + 0.1
        SendSM "read?", stat%
        ReadSM r$
        Curr = Val(Mid(r$, 15, 13)) * 1000
        If Curr > Val(txtCurrentLimit.Text) * 98 / 100 Then
            Exit Do
        End If
        txtVEnd = i
        If sldVset.Max < i * 100 Then
            sldVset.Max = i * 100
            sldVset.TickFrequency = (sldVset.Max - sldVset.Min) / 100
        End If
        sldVset.value = i * 100
        DoEvents
    Loop While i < 20 And optFindV.value = True
    sldVset.Max = i * 100
'********************************************
    i = 0
    Do
        VLevel$ = ":SOUR:VOLT:LEV " & Str(i)
        SendSM VLevel$, stat%
        i = i - 0.1
        SendSM "read?", stat%
        ReadSM r$
        Curr = Val(Mid(r$, 15, 13)) * 1000
        If Curr < -Val(txtCurrentLimit.Text) * 98 / 100 Then
            Exit Do
        End If
        txtVStart = i
        If sldVset.Min > i * 100 Then
            sldVset.Min = i * 100
            sldVset.TickFrequency = (sldVset.Max - sldVset.Min) / 100
        End If
        sldVset.value = i * 100
```

```
        DoEvents
    Loop While i > -20 And optFindV.value = True
    sldVset.Min = i * 100
    SendSM ":SOUR:VOLT:LEV 0.0", stat%
    sldVset.value = 0
    optFindV.value = False
End Sub

Private Sub optMidGND_Click()
    If optMidGND.value = True Then
        InitializeSW
        SendSW ":clos (@ 1!22)", stat%
        SendSW ":open (@ 1!21)", stat%
        fMainForm.chkMidGND.value = 1
    End If
End Sub

Private Sub optSepGND_Click()
    If optSepGND.value = True Then
        InitializeSW
        SendSW ":clos (@ 1!21)", stat%
        SendSW ":open (@ 1!22)", stat%
        fMainForm.chkMidGND.value = 0

    End If
End Sub

Private Sub optStop_Click()
    optStop.value = False
End Sub

Private Sub sldVset_Change()
    VLevel$ = ":SOUR:VOLT:LEV " & sldVset.value / 100
    SendSM VLevel$, stat%
End Sub

Private Sub TgB_Click(Index As Integer)
    InitializeSW
    If TgB(Index).value = True Then
        lblLED(Index).BackColor = &HFF00&
        fMainForm.PushChannel Index + 1
        DoEvents

        For i = 0 To 9
            If i - Index <> 0 Then TgB(i).value = False
        Next i
        CloseSW Index + 1, optMidGND
    Else
        lblLED(Index).BackColor = &H0&
        CloseSW 0, optMidGND
    End If
End Sub

Private Sub Timer1_Timer()
    SendSM "read?", stat%
    ReadSM r$
    lblV.Caption = Format(Val(Mid(r$, 1, 13)), "#0.00000")
```

```
    lblI.Caption = Format(Val(Mid(r$, 15, 13)) * 1000000, "###0.00000")
End Sub

Private Sub txtCurrentlimit_Change()
    If txtCurrentLimit <> "" Then
        If txtCurrentLimit > 10 Then
            cho = MsgBox("! WARNING ! Your Current Limit is grater than 10mA It
may be harmfull for your divice" & Chr(10) & cr & "Are you sure ?",
vbExclamation + vbYesNo, "WARNING")
            If cho <> 6 Then
                txtCurrentLimit = 10
            End If
        End If
        ILimit = Val(txtCurrentLimit.Text)
        iprot$ = ":SENS:CURR:PROT " & Str(ILimit / 1000)
        SendSM iprot$, stat%
    End If


End Sub

Private Sub txtVEnd_Change()
    If Val(txtVEnd) > sldVset.Min / 100 And Val(txtVEnd) < 20 Then
        sldVset.Max = Val(txtVEnd) * 100
        sldVset.TickFrequency = (sldVset.Max - sldVset.Min) / 100
    End If
End Sub

Private Sub txtVStart_Change()
    If Val(txtVStart) < sldVset.Max / 100 And Val(txtVStart) > -20 Then
        sldVset.Min = Val(txtVStart) * 100
        sldVset.TickFrequency = (sldVset.Max - sldVset.Min) / 100
    End If
End Sub



***********************************************************************
frmTempvsTime.frm
***********************************************************************
'Copyrights - Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State
University


Dim oBook As Excel.Workbook
Dim oSheet As Excel.Worksheet
Dim oChart As Excel.Chart
Dim StartTime
Dim FR%
Dim LT1, LT2 As Single


Private Sub cmdClose_Click()
    Unload Me
End Sub
```

```
Private Sub cmdMaximize_Click()
    Me.WindowState = 2
End Sub

Private Sub cmdMinimize_Click()
    Me.WindowState = 1
End Sub

Private Sub cmdNormal_Click()
    Me.WindowState = 0
End Sub

Private Sub cmdSaveAs_Click()
    SaveAs
End Sub

Private Sub Form_Load()
'   OLE1.CreateEmbed "", "excel.chart"
    Set oBook = OLE1.object
    Set oChart = oBook.Charts(1)
    Set oSheet = oBook.Worksheets(1)
    StartTime = Time()
Exit Sub
    oSheet.Cells.Clear
    oChart.ChartType = xlXYScatterLines
    oChart.SetSourceData oSheet.Range("A13:B114"), xlColumns
    With oChart
        .HasTitle = True
        .ChartTitle.Text = "IV Curve of the Sample"
        .HasLegend = True
        .Legend.Interior.ColorIndex = xlNone
        .Legend.Border.LineStyle = xlNone
        .Legend.Font.Size = 10
        .Legend.Left = 1
        .Legend.Top = 20
        .Axes(xlCategory, xlPrimary).HasTitle = True
        .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "Voltage (V)"
        .Axes(xlValue, xlPrimary).HasTitle = True
        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Current Density
(A/m²)"
    End With
    With oChart.Axes(xlValue)
        .HasMajorGridlines = False
        .HasMinorGridlines = False
        .MajorTickMark = xlInside
        .MinorTickMark = xlNone
        .TickLabelPosition = xlNextToAxis
'        .TickLabels.NumberFormat = "0.00"
        .TickLabels.Font.Size = 10
        .ScaleType = xlLogarithmic
    End With
    With oChart.Axes(xlCategory)
        .HasMajorGridlines = False
        .HasMinorGridlines = False
        .MajorTickMark = xlInside
        .MinorTickMark = xlNone
        .TickLabelPosition = xlNextToAxis
```

```
 '      .TickLabels.NumberFormat = "0.00"
        .TickLabels.Font.Size = 10
    End With

    oChart.PlotArea.Interior.ColorIndex = xlNone

    OLE1.Visible = True
 '   OLE1.Close  'Deactivate the OLE container
    Application.Assistant.Visible = True

End Sub

Private Sub Form_Resize()
    On Error Resume Next
    OLE1.Move 200, 200, Me.ScaleWidth - 400, Me.ScaleHeight - 1000
    Frame1.Top = Me.ScaleHeight - 800
    Frame1.Left = ((Me.ScaleWidth - 4320) / 2)
End Sub

Sub Save()
    Dim sFile As String
    On Error GoTo ErrHandler
    sFile = frmSetup.txtProjectName & "_TempVsTime.xls"
    oBook.Windows.Item(1).Visible = True
    oBook.Windows.Item(1).WindowState = xlMaximized
    oBook.SaveCopyAs sFile
    Caption = "T vs. t  |" & sFile
    cmdClose.Enabled = False
ErrHandler:
    Exit Sub
End Sub

Sub SaveAs()
    Dim sFile As String
    On Error GoTo ErrHandler
    With CommonDialog
        .CancelError = True
        .Filter = "Excel File (*.xls)|*.xls"
        .DialogTitle = "Save IV Curve File As"
        .Flags = cdlOFNOverwritePrompt
        .ShowSave
        If Len(.filename) = 0 Then
            Exit Sub
        End If
        sFile = .filename
    End With
    oBook.Windows.Item(1).Visible = True
    oBook.Windows.Item(1).WindowState = xlMaximized
    oBook.SaveCopyAs sFile
    Caption = "T vs. t  |" & sFile

ErrHandler:
    Exit Sub
End Sub

Sub FillData(T1 As Single, T2 As Single, IVPoint As Boolean)
    With oSheet
```

```
        If  Abs(T1  -  LT1)  >  frmOptions.txtTempTol  Or  Abs(T2  -  LT2)  >
frmOptions.txtTempTol Then
            RW% = .Cells(14, 3).value + 20
           .Cells(14, 3).value = RW% - 19
            If RW% > 20 Then
                LT1 = T1
                LT2 = T2
            End If
        Else
            RW% = .Cells(14, 3).value + 19
        End If
        .Cells(RW%, 1) = Time - StartTime
        .Cells(RW%, 2) = T1
        .Cells(RW%, 3) = T2
        oChart.SeriesCollection(1).XValues = .Range("A20:A" & RW%)
        oChart.SeriesCollection(1).Values = .Range("B20:B" & RW%)
        oChart.SeriesCollection(2).XValues = .Range("A20:A" & RW%)
        oChart.SeriesCollection(2).Values = .Range("C20:C" & RW%)
        If IVPoint = True Then
            If FR% = 0 Then FR% = RW%
            .Cells(RW%, 4) = T2 + 2 * a
            oChart.SeriesCollection(3).XValues = .Range("A" & FR% & ":A" & RW%)
            oChart.SeriesCollection(3).Values = .Range("D" & FR% & ":D" & RW%)
        End If
    End With
End Sub

Sub OldSetTemp(t As Single)
    Dim T1 As Single
    Dim T2 As Single
    Dim H As Single
    fMainForm.tmrTemp.Enabled = False
    cr = Chr(13)
    '************* To find Heter ON/OFF and set it to ON
    SendTC "S3000" & cr, stat%
    SendTC "X" & cr, stat%
    Wait 1
    If Val(Mid(ReadTC("H" & cr), 2, 5)) = 0 Then
        SendTC "X" & cr, stat%
    End If
    SendTC "S100" & cr, stat%

    '***************
    SendTC "S" & t * 10 & cr, stat%
    T1 = Val(Mid(ReadTC("T" & cr), 2, 5))
    T2 = Val(Mid(ReadTC("t" & cr), 2, 5))
    H = Val(Mid(ReadTC("H" & cr), 2, 5))
    fMainForm.UpdateTemp T1, T2, H

    TSet = t
    While Abs(T2 - t) > frmOptions.txtTempTol
        While Abs(T1 - TSet) > Val(frmOptions.txtTempTol)
            '************* Check START/PAUSE/STOP Status*************
            With fMainForm
                If .tbToolBar.Buttons.Item(12).value = tbrUnpressed Then
                    If .tbToolBar.Buttons.Item(13).value = tbrPressed Then
                        '.MousePointer = 0
```

```
                    While .tbToolBar.Buttons.Item(13).value = tbrPressed
                        DoEvents
                    Wend
                End If
                If .tbToolBar.Buttons.Item(12).value = tbrUnpressed Then
                    '.MousePointer = 0
                    Exit Sub
                End If
            End If
            '.MousePointer = 11
        End With
        '**********************************************************
        Wait frmOptions.txtDelayT
        T1 = Val(Mid(ReadTC("T" & cr), 2, 5))
        T2 = Val(Mid(ReadTC("t" & cr), 2, 5))
        H = Val(Mid(ReadTC("H" & cr), 2, 5))
        fMainForm.UpdateTemp T1, T2, H
        FillData T1, T2, False
        DoEvents
    Wend
    TSet = Val(Mid(ReadTC("S" & cr), 2, 5)) - (frmOptions.txtTempTol / 2)
    Wait 2
    SendTC "S" & TSet * 10 & cr, stat%
Wend
FillData T1, T2, True
End Sub

Sub SetTemp(t As Single)
    Dim T1 As Single
    Dim T2 As Single
    Dim H As Single
    fMainForm.tmrTemp.Enabled = False
    cr = Chr(13)
    Dim GCount As Integer

    '************* To find Heter ON/OFF and set it to ON
    SendTC "S3000" & cr, stat%
    SendTC "X" & cr, stat%
    Wait 1
    If Val(Mid(ReadTC("H" & cr), 2, 5)) = 0 Then
        SendTC "X" & cr, stat%
    End If
    SendTC "S100" & cr, stat%

    '***************
    SendTC "S" & t * 10 & cr, stat%
    T1 = Val(Mid(ReadTC("T" & cr), 2, 5))
    T2 = Val(Mid(ReadTC("t" & cr), 2, 5))
    H = Val(Mid(ReadTC("H" & cr), 2, 5))
    fMainForm.UpdateTemp T1, T2, H

    n% = 0
    GCount = 0
    While   GCount   <   20   /   frmOptions.txtDelayT   And   n%   <   (1000   /
frmOptions.txtDelayT)
        n = n + 1
        '************* Check START/PAUSE/STOP Status*************
```

```
        With fMainForm
            If .tbToolBar.Buttons.Item(12).value = tbrUnpressed Then
                If .tbToolBar.Buttons.Item(13).value = tbrPressed Then
                    '.MousePointer = 0
                    While .tbToolBar.Buttons.Item(13).value = tbrPressed
                        DoEvents
                    Wend
                End If
                If .tbToolBar.Buttons.Item(12).value = tbrUnpressed Then
                    '.MousePointer = 0
                    Exit Sub
                End If
            End If
            '.MousePointer = 11
        End With
        '**********************************************************
        Wait frmOptions.txtDelayT
        T1 = Val(Mid(ReadTC("T" & cr), 2, 5))
        T2 = Val(Mid(ReadTC("t" & cr), 2, 5))
        H = Val(Mid(ReadTC("H" & cr), 2, 5))
        fMainForm.UpdateTemp T1, T2, H
        FillData T1, T2, False
        DoEvents

        If Abs(T1 - t) <= Val(frmOptions.txtTempTol) Then
            GCount = GCount + 1
        End If
    Wend
    FillData T1, T2, True
End Sub



***********************************************************************
TempController.frm
***********************************************************************
'Copyrights - Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State
University

Private cr As String

Private Sub cmdApply_Click()
    SendTC "S" & txtSetPoint.Text * 10 & cr, stat%


    txtSetPoint.Text = Format(Mid(ReadTC("S" & cr), 2, 5), "000.0")
    txtProTerm.Text = Mid(ReadTC("P" & cr), 2, 2)
    txtIntTerm.Text = Mid(ReadTC("I" & cr), 2, 2)
    txtDerTerm.Text = Mid(ReadTC("D" & cr), 2, 2)
End Sub

Private Sub cmdExit_Click()
    Timer1.Enabled = False
    Unload Me
End Sub

Private Sub cmdHeater_Click()
```

```
    SendTC "X" & cr, stat%
    If lblLED.BackColor = &H0& Then
        lblLED.BackColor = &HFF00&
    Else
        lblLED.BackColor = &H0&
    End If
        MousePointer = 11
        Sleep 1000
    If Val(Mid(ReadTC("H" & cr), 2, 5)) > 0 Then
        lblLED.BackColor = &HFF00&
    End If
        MousePointer = 0

End Sub

Private Sub Form_Load()
    cr = Chr(13)
    txtSetPoint.Text = Format(Mid(ReadTC("S" & cr), 2, 5), "000.0")
    txtProTerm.Text = Mid(ReadTC("P" & cr), 2, 2)
    txtIntTerm.Text = Mid(ReadTC("I" & cr), 2, 2)
    txtDerTerm.Text = Mid(ReadTC("D" & cr), 2, 2)
    Timer1.Enabled = True
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Timer1.Enabled = False
'     ilonl Dev%, 0
End Sub

Private Sub Timer1_Timer()
    lblT1.Caption = Format(Mid(ReadTC("T" & cr), 2, 5), "000.0 K")
    lblT2.Caption = Format(Mid(ReadTC("t" & cr), 2, 5), "000.0 K")
    lblH.Caption = Format(Mid(ReadTC("H" & cr), 2, 5), "000.0 W")
End Sub



***********************************************************************
Keithley7001.frm
***********************************************************************
'Copyrights - Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State
University

Private Sub optMidGND_Click()
    If optMidGND.value = True Then
        InitializeSW
        SendSW ":clos (@ 1!22)", stat%
        SendSW ":open (@ 1!21)", stat%
    End If
End Sub

Private Sub optSepGND_Click()
    If optSepGND.value = True Then
        InitializeSW
        SendSW ":clos (@ 1!21)", stat%
        SendSW ":open (@ 1!22)", stat%
    End If
```

```
End Sub

Private Sub TgB_Click(Index As Integer)
    InitializeSW
    If TgB(Index).value = True Then
        lblLED(Index).BackColor = &HFF00&
        DoEvents
        For i = 0 To 9
            If i - Index <> 0 Then TgB(i).value = False
        Next i
        CloseSW Index + 1, optMidGND
    Else
        lblLED(Index).BackColor = &H0&
        CloseSW 0, optMidGND
    End If
End Sub

**********************************************************************
frmDataCollecting.frm
**********************************************************************
'Copyrights - Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State
University


Dim DFileName(1 To 10) As String
Dim mesaSize(1 To 10) As Single
Dim NoChannel As Integer
Dim fNo1 As Integer
Dim fNo2 As Integer
Dim tStart As Variant

Private Sub Form_Load()
optPause.value = False
optPause.Enabled = False
optStop.Enabled = False
optPlay.value = False

End Sub

Private Sub optPlay_Click()
    If optPause.Enabled = False And optStop.Enabled = False Then
        optPause.Enabled = True
        optStop.Enabled = True
    End If
    fNo1 = FreeFile()
    Open Left(DFileName(1), Len(DFileName(1) - 2)) & "_" & "Tvst" & ".txt" For
Output As #fNo1

    SetSourceMeter
    tStart = Timer()
    Dim Temp As Single
    pbrOP.value = 2

    For Temp = frmSetup.txtStartT To frmSetup.txtEndT Step frmSetup.txtStepT
        lblStatus.Caption = "Waiting for Temp. " & Temp
        SetTemp (Temp)  'To set the temp
```

```
        For Channel = 1 To NoChannel
           GetIV Channel, Temp
           pbrOP.value    =    pbrOP.value    +    Int(((1    /    NoChannel)    *
(frmSetup.txtStepT / Abs(frmSetup.txtEndT - frmSetup.txtStartT))))
           Close #fNo1
           fNo1 = FreeFile()
           Open  Left(DFileName(1),  Len(DFileName(1)  -  2))  &  "_"  &  "Tvst"  &
".txt" For Append As #fNo1
        Next Channel

        pbrOP.value   =   Int(Abs(Temp   -   frmSetup.txtStartT)   *   100   /
Abs(frmSetup.txtEndT - frmSetup.txtStartT))

    Next Temp

    Close fNo1
End Sub

Private Sub optStop_Click()
optStop.value = False
End Sub

Sub IVFNames(n As Integer, FName() As String, MSize() As Single)
NoChannel = n
For i = 1 To n
    DFileName(i) = Left(FName(i), Len(FName(i)) - 3)
    mesaSize(i) = MSize(i)
Next i
End Sub

Sub SetTemp(t As Single)
    Dim T1 As Single
    Dim T2 As Single
    Dim H As Single
    fMainForm.tmrTemp.Enabled = False
    cr = Chr(13)
    Dim GCount As Integer

    '************* To find Heter ON/OFF and set it to ON
    SendTC "S3000" & cr, stat%
    SendTC "X" & cr, stat%
    Wait 1
    If Val(Mid(ReadTC("H" & cr), 2, 5)) = 0 Then
        SendTC "X" & cr, stat%
    End If
    SendTC "S100" & cr, stat%

    '***************
    SendTC "S" & t * 10 & cr, stat%
    T1 = Val(Mid(ReadTC("T" & cr), 2, 5))
    T2 = Val(Mid(ReadTC("t" & cr), 2, 5))
    H = Val(Mid(ReadTC("H" & cr), 2, 5))
    fMainForm.UpdateTemp T1, T2, H

    n% = 0
    GCount = 0
```

```
    While   GCount   <   20   /   frmOptions.txtDelayT   And   n%   <   (1000   /
frmOptions.txtDelayT)
        n = n + 1
        '************ Check START/PAUSE/STOP Status************
            If optPlay.value = False Then
                If optPause.value = True Then
                    MousePointer = 0
                    While optPause.value = True
                        DoEvents
                    Wend
                    MousePointer = 11
                End If
                If optPlay.value = False Then
                    MousePointer = 0
                    Exit Sub
                End If
            End If
        '****************************************************
        Wait frmOptions.txtDelayT
        T1 = Val(Mid(ReadTC("T" & cr), 2, 5))
        T2 = Val(Mid(ReadTC("t" & cr), 2, 5))
        H = Val(Mid(ReadTC("H" & cr), 2, 5))
        fMainForm.UpdateTemp T1, T2, H
        'FillData T1, T2, False
        DoEvents

        If Abs(T1 - t) <= Val(frmOptions.txtTempTol) Then
            GCount = GCount + 1
        End If
    Wend
    'FillData T1, T2, True
End Sub

Sub GetIV(Channel As Integer, Temp As Single)
    With frmSetup
        fNo2 = FreeFile()
        Open DFileName(Channel) & "_" & Str(Temp) & ".txt" For Output As #fNo2
        CNo% = Asc(.TabStrip1.Tabs(Channel).Caption) - 64
        CloseSW CNo%, frmSetup.optMidGND
        lblStatus.Caption    =    "Taking    IV    data    for    ..."    &
Right(DFileName(Channel), 10) & "_" & Str(Temp) & ".txt"

        T2 = Val(Mid(ReadTC("t" & cr), 2, 5))
        Print #fNo1, Int(Timer() - tStart); " , "; T2; "  Starting point for
Channel:"; Channel

        Print #fNo2, "Viraj2005 Automated IVT Setup Data File"
        Print #fNo2, "Sample Name:, "; DFileName(Channel)
        Print #fNo2, "Description:"
        Print #fNo2, "No of Graphs & Channel:"
        Print #fNo2, "Date & Time:, "; Date; ", "; Time
        Print #fNo2, "Mesa; Area(mm²):"; mesaSize(Channel)
        Print #fNo2, "Start; V(V):, "; .txtStartV
        Print #fNo2, "End V (V)  :, "; .txtEndV
        Print #fNo2, "Step V(V):, "; .txtStepV
        Print #fNo2, "Current limit(mA):, "; .txtCurrentLimit
        Print #fNo2, "Delay:, "; .txtDelay
```

```
        Print #fNo2, "Avg.:, "; .txtAvg
        Print #fNo2, "No of Cycles:, "; .txtNoCycle
        Print #fNo2, "No of Data:, "; ((.txtEndV - .txtStartV) / .txtStepV) + 1
        Print #fNo2, "Start Temp(K):, "; .txtStartT
        Print #fNo2, "End Temp (K):, "; .txtEndT
        Print #fNo2, "Step Temp(K):, "; .txtStepT
        Print #fNo2, "Sample Temp(K):, "; Temp
        Print #fNo2, "Voltage (V)" & " , " & "Current (A)"

        Dim V As Single
test:
        SendSM "*RST", stat%
        If stat% <> 0 Then
           cho = MsgBox("KEITHLEY 2400 SourceMeter not found", vbRetryCancel +
vbExclamation, "GPIB Address Error")
           If cho = 4 Then
                GoTo test
           Else
               Exit Sub
           End If
        End If

        SendSM ":OUTPUT ON", stat%
        StartV = Val(.txtStartV)
        EndV = Val(.txtEndV)
        StepV = Val(.txtStepV)
        NoCycle% = CInt(Val(.txtNoCycle))
        ILimit = Val(.txtCurrentLimit) / 1000

        For NoSweep% = 1 To NoCycle%

            For V = StartV To EndV Step StepV
                t = Timer()
                ok = 0
                SendSM ":SOUR:VOLT:LEV " & Str(V), stat%

 '******************** If Gradient Check ON ***********************
                If chkGradiCheck.value = 1 Then

                    While ok = 0
                        II = 0
                        SendSM "read?", stat%
                        ReadSM r$
                        IOld = Val(Mid(r$, 15, 13))
                        n = 0

                        For j = 1 To .txtAvg.Text
 '************* Check START/PAUSE/STOP Status*************
                            If optPlay.value = False Then
                                If optPause.value = True Then
                                    MousePointer = 0
                                    While optPause.value = True
                                        DoEvents
                                    Wend
                                    MousePointer = 11
                                End If
                                If optPlay.value = False Then
```

```
                    MousePointer = 0
                    Close #fNo2
                    Exit Sub
                End If
            End If

'**********************************************************

            SendSM "read?", stat%
            ReadSM r$
            INew = Val(Mid(r$, 15, 13))

            If IOld < INew Then n = n + 1
            If IOld > INew Then n = n - 1
            II = II + INew
            IOld = INew
            DoEvents
        Next j

        If Abs(n) < 4 Then ok = 1
    Wend
Else

'******************** If Gradient Check OFF *************************
    II = 0
    For j = 1 To Val(txtAvg.Text)
'************* Check START/PAUSE/STOP Status*************
        If optPlay.value = False Then
            If optPause.value = True Then
                MousePointer = 0
                While optPause.value = True
                    DoEvents
                Wend
                MousePointer = 11
            End If
            If optPlay.value = False Then
                MousePointer = 0
                Close #fNo2
                Exit Sub
            End If
        End If

'**********************************************************
        SendSM "read?", stat%
        ReadSM r$
        INew = Val(Mid(r$, 15, 13))
        II = II + INew
        DoEvents
    Next j

'*****************************************************************
End If
VV = Val(Mid(r$, 1, 13))
II = II / txtAvg.Text
Print #fNo2, V; " , "; II
INew = II
```

```
'*************** If Current > comp. limit find edge ****************
                If Abs(INew) > (0.98 * ILimit) Then
                    If Abs(V + StepV) < Abs(V) Then  'Starting edge correction
                        V0 = V
                        V1 = 0
                        While Abs(INew) > (0.98 * ILimit) Or Abs(INew) < (0.97
* ILimit)
                            V2 = (V0 - V1) / 2
                            SendSM ":SOUR:VOLT:LEV " & Str(V2), stat%
                            SendSM "read?", stat%
                            ReadSM r$
                            INew = Val(Mid(r$, 15, 13))
                            If Abs(INew) > (0.98 * ILimit) Then
                                V0 = V2
                            Else
                                V1 = V2
                            End If
                        Wend
                        For FV = V + StepV To V2 Step StepV
                            Print #fNo2, FV; " , "; II
                        Next FV
                        V = FV + StepV

                    Else  'Ending edge correction
                        For FV = V + StepV To EndV Step StepV
                            Print #fNo2, FV; " , "; II
                        Next FV
                        Exit For
                    End If
                End If

'*******************************************************************
                II = 0
                pbrCIV.value = Int(Abs(V - StartV) * 100 / Abs(EndV - StartV))

                If (V - StartV) Mod 10 * StepV = 0 Then
                    T2 = Val(Mid(ReadTC("t" & cr), 2, 5))
                    Print #fNo1, Int(Timer() - tStart); " , "; T2
                End If
            Next V

            TempStartV = StartV
            StartV = EndV
            EndV = TempStartV
            StepV = -StepV
        Next NoSweep%
    End With
    T2 = Val(Mid(ReadTC("t" & cr), 2, 5))
    Print #fNo1, Int(Timer() - tStart); " , "; T2; "  End point for Channel:";
Channel


    Close #fNo2
 End Sub

Sub SetSourceMeter()
    With frmSetup
```

```
        Iprot$ = ":SENS:CURR:PROT " & Str(.txtCurrentLimit.Text / 1000)
        VLevel$ = ":SOUR:VOLT:LEV " & .txtStartV.Text
        SDelay$ = ":SOUR:DEL " & .txtDelay.Text
        SendSM ":SOUR:FUNC VOLT", stat%
        SendSM ":SOUR:VOLT:MODE FIXED", stat%
        SendSM ":SENS:FUNC 'CURR'", stat%
        SendSM Iprot$, stat%
        SendSM VLevel$, stat%
        SendSM ":SENS:CURR:RANG:AUTO ON", stat%
        SendSM SDelay$, stat%
    End With
End Sub
```

B.2 UV-VIS-NIR detector spectral response characterization software

The UV, VIS and NIR detector characterization system is controlled using an in-house software package developed with Microsoft Visual Basic. The heart of this setup is the Spectral Products DK480 monochromator with 3 gratings, a motorized filter wheel, motorized entrance and exit slits. A schematic diagram of the complete setup is shown in Figure B.12. The motorized mirror is used to select the light source: Deuterium UV lamp, tungsten halogen VIS-NIR or HeNe laser. A multi-sample detector stage has two calibrated detectors: UV enhanced Si and InGaAs photodetectors. Another four detector positions are available for custom detector characterization. Software features include automatic output light intensity calibration, real time detector spectral response calculation, auto-align detectors for maximum energy, fully automatic light source, grating and filter selection. The DK480 monochromator and SR850 lock-in amplifier are controled through RS232 and GPIB interfaces, respectively. Mirror and stage stepper motor controllers, shutter, and power unit functions are controlled through the PCI8255 general-purpose PCI board.

B.2.1 UV-VIS-NIR setup instrument configuration

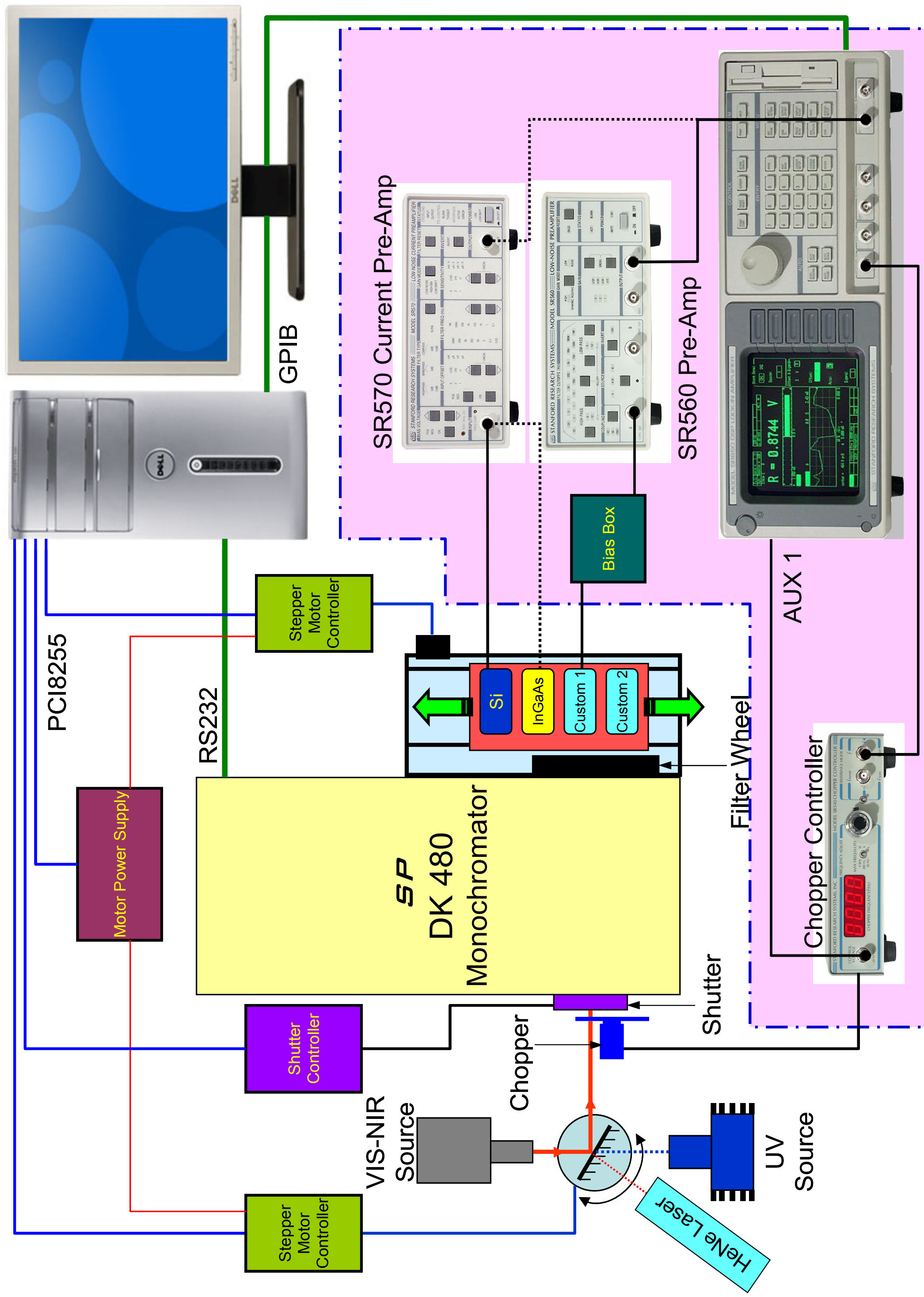


Figure B.12 Block diagram of the UV-VIS-NIR detector characterization system.

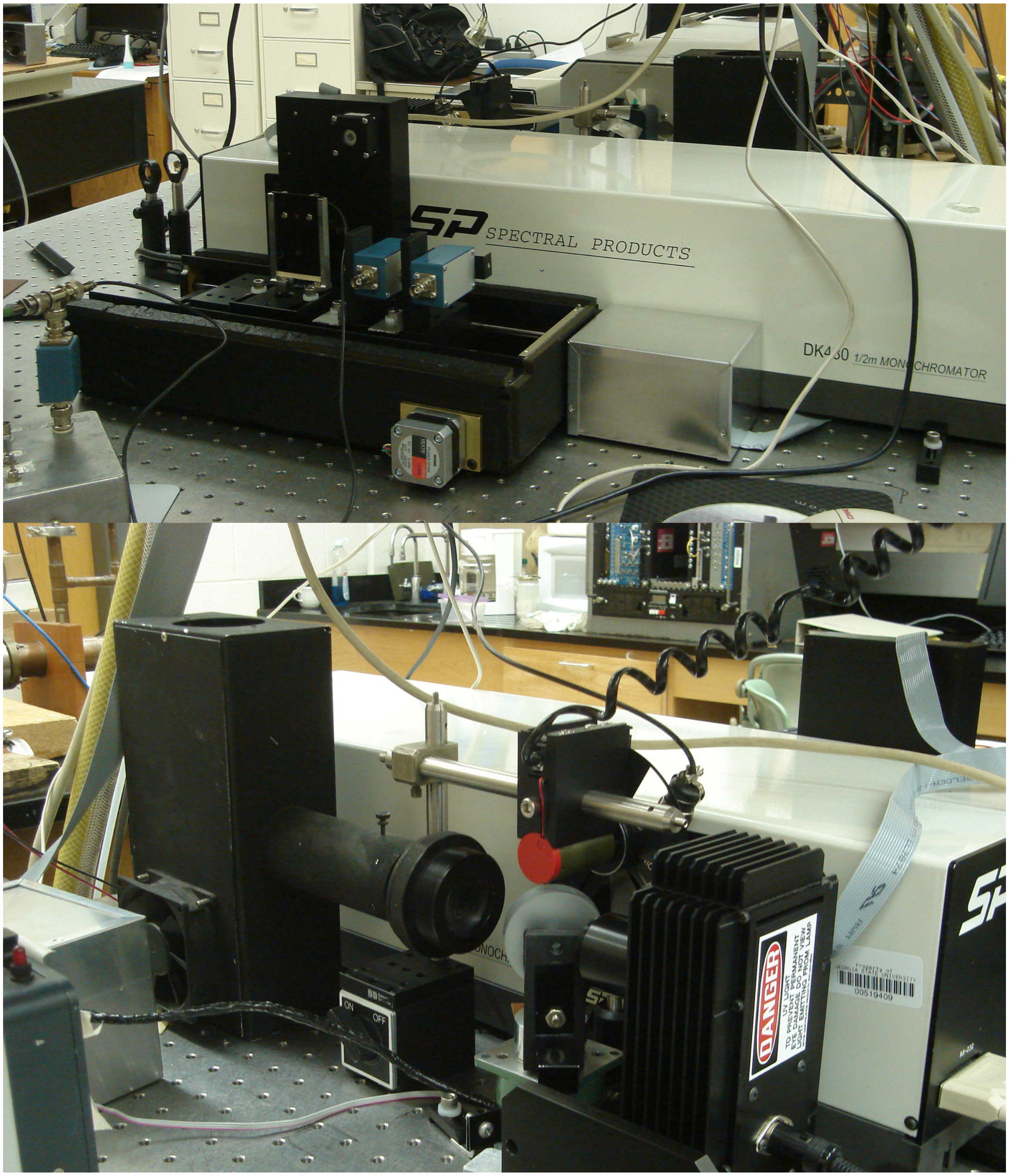


Figure B.13 Photograph of the UV-VIS-NIR detector characterization system.

B.2.2 User interfaces

Figure B.14 Hardware initialization window. At the starting step of the program, it will search availability of all the instruments and communication status, power up the stepper motor controllers, find the zero position for the stepper motor controlled mirror and sample-stage, and check the shutter status. If successful, this window will automatically disappear, otherwise it shows an error message asking the user to check the status of the particular instrument.

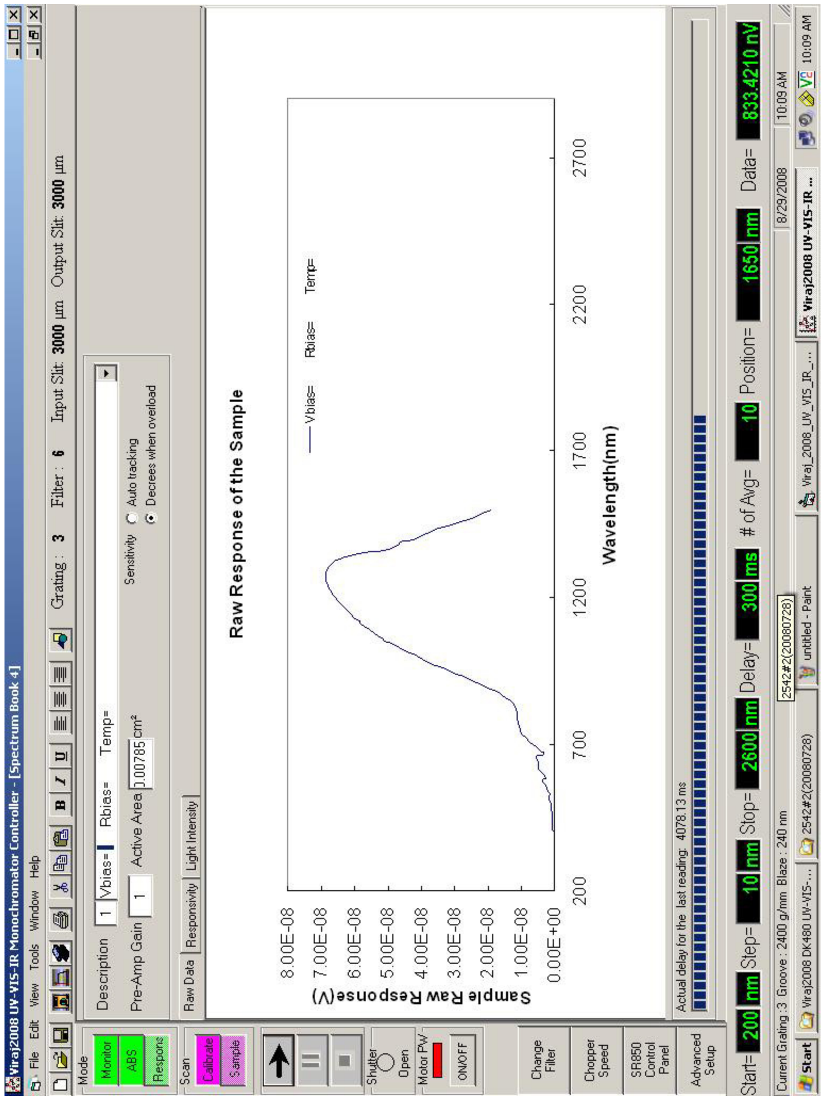


Figure B.15 The main user interface of the UV-VIS-NIR detector characterization software. There are three different user selectable working modes: (i) regular spectrometer mode for absorption and reflection measurements, (ii) light intensity calibration mode, and (iii) custom detector characterization mode. The user can observe the real time raw spectrum, responsivity (V/W), and incident light intensity for the custom detectors.

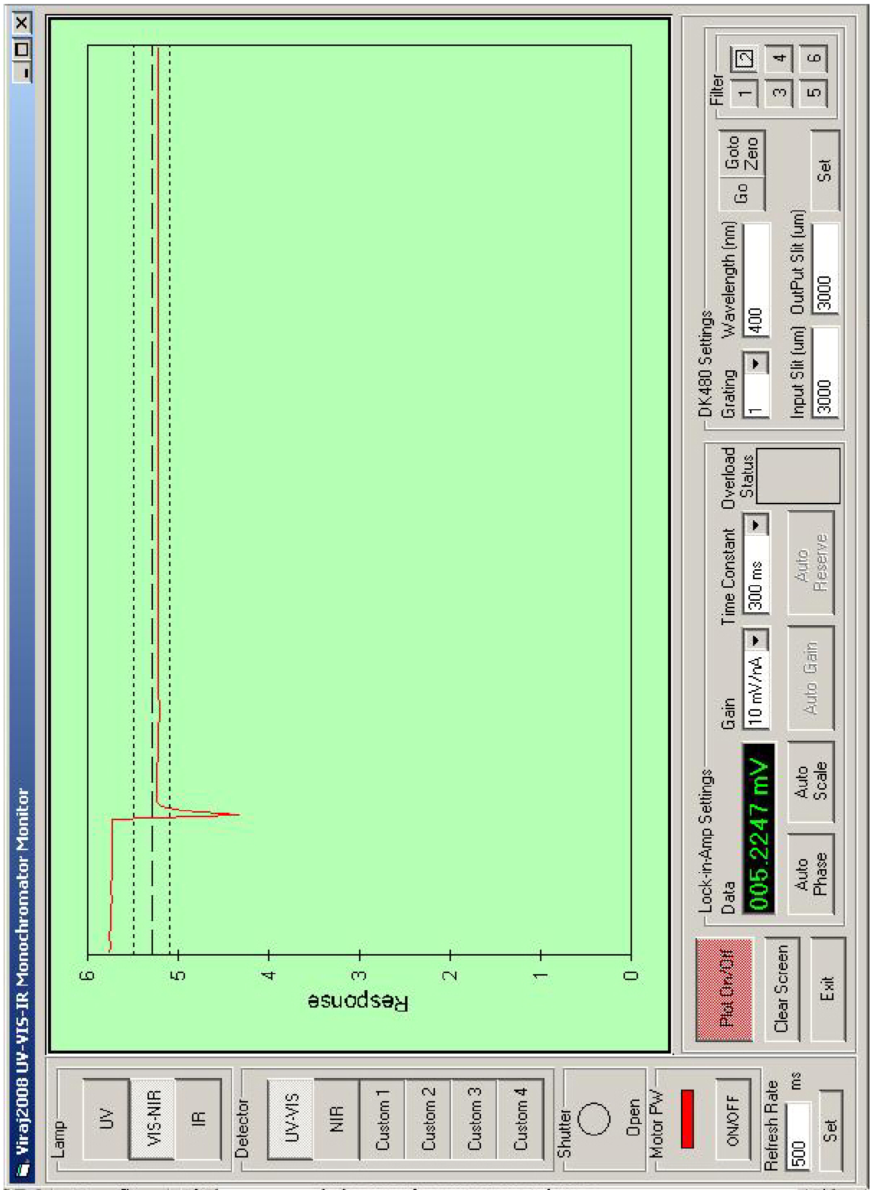


Figure B.16 The “monitor-mode” control panel allows the user to control all the instruments, motors and stages manually through the software. The user can change the lock-in-amplifier and monochromator parameters, properly align the detectors, and real time observe the output of the selected detector at a selected wavelength, upon the selection of a light source and a filter.

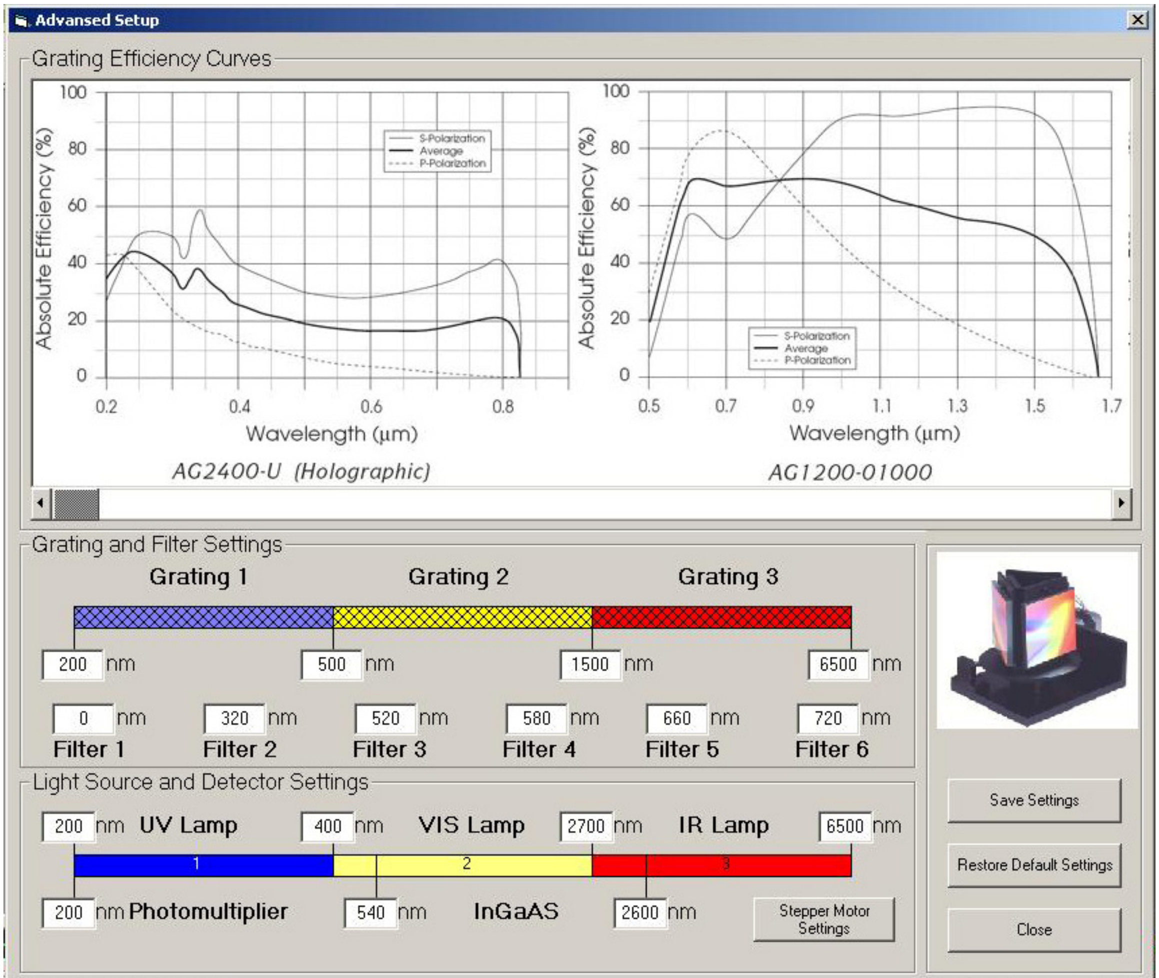


Figure B.17 The "Advanced setup" control panel allows the user to change the hardware configurations such as wavelength range for each grating, wavelength position for each filter, wavelength ranges for each light source and the standard detectors (Si and InGaAs). Grating efficiency curves for all available gratings are inserted on the top of the panel so that user can easily determine which grating is most suitable in the wavelength region of interest. Users can save their own configuration settings, and they can also restore the original settings by pressing the "Restore Default Settings" button.

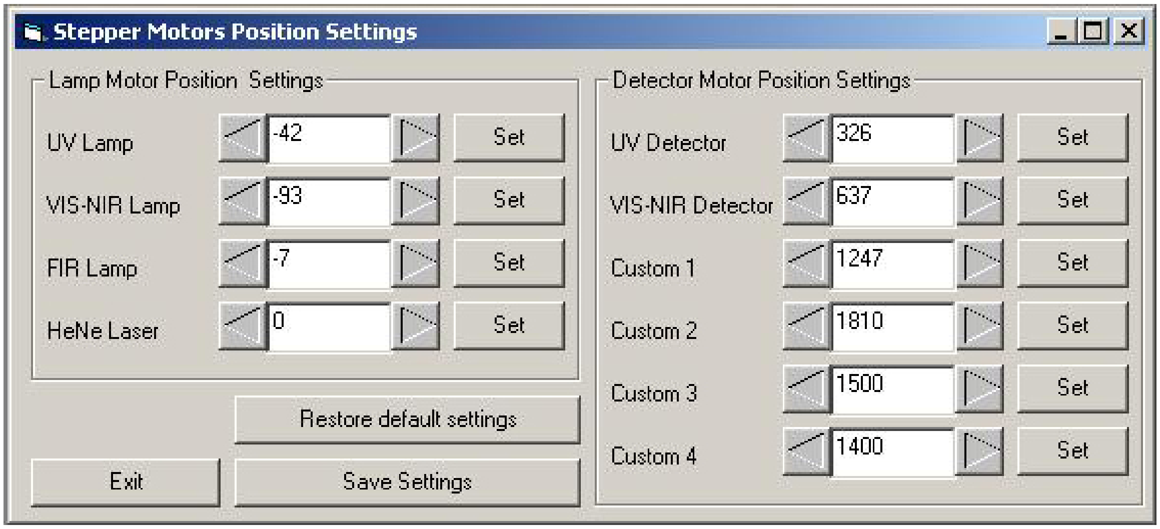


Figure B.18 A view of the "stepper motor position settings". This is another advanced user interface that allows the user to set the light-source-selecting-mirror positions and detector mounting stage positions. The numbers represent the actual number of steps from the zero position of the each stepper- motor. After a hardware modification, such as adding a new sample mount to the custom detector slot, the user can configure the new position and save it for future use.

Figure B.19 The chopper control panel allows the user to set the chopper frequency to one of the preset values or any other custom value. The user can observe the current frequency on the screen.

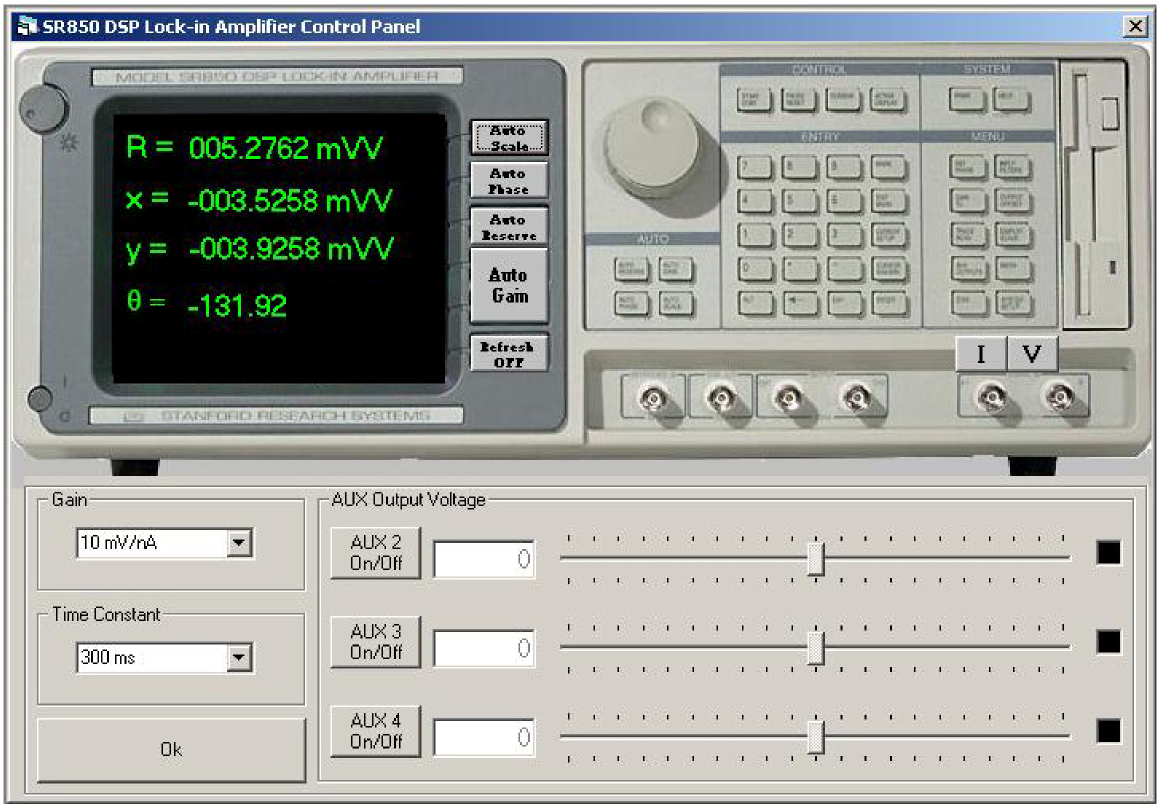
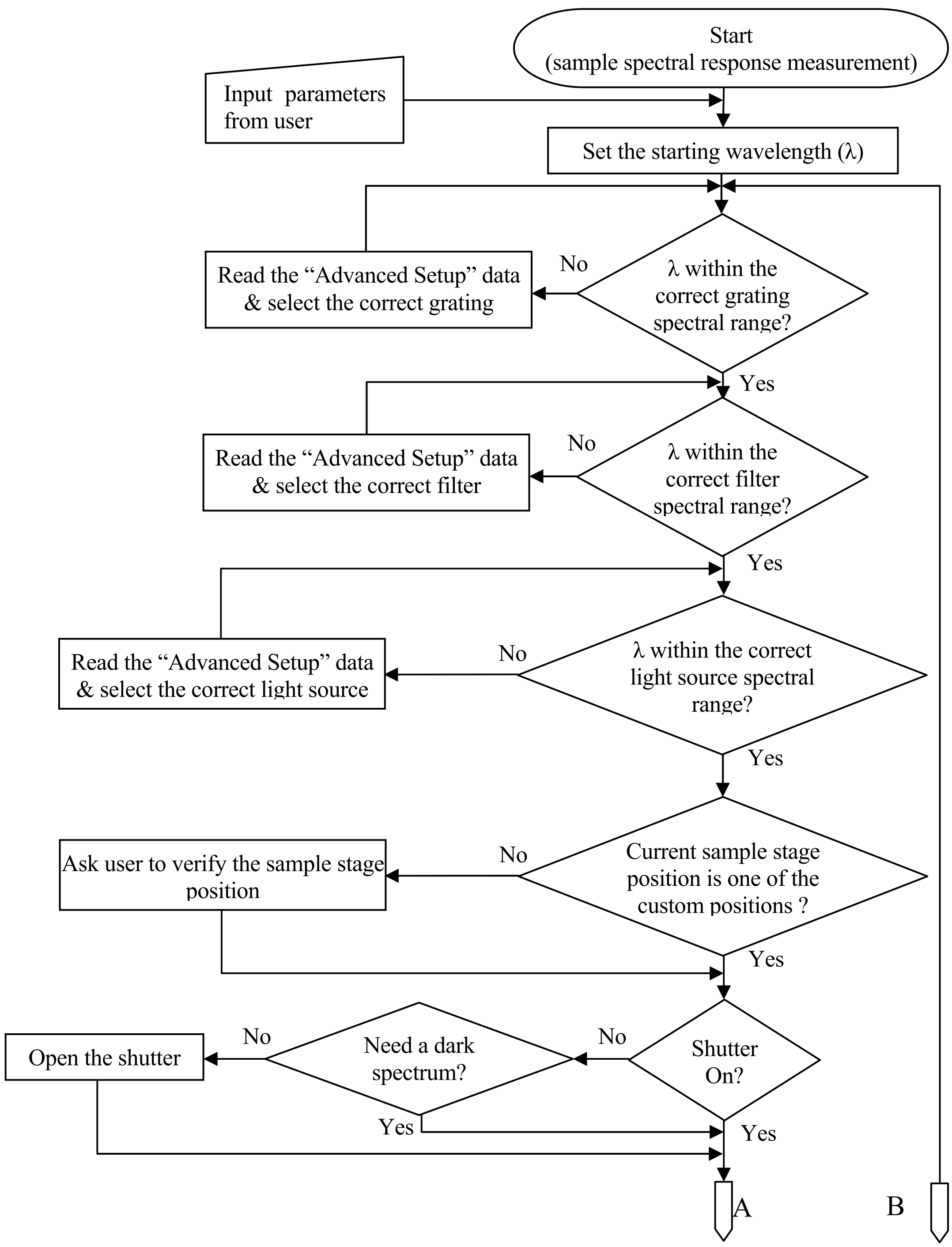
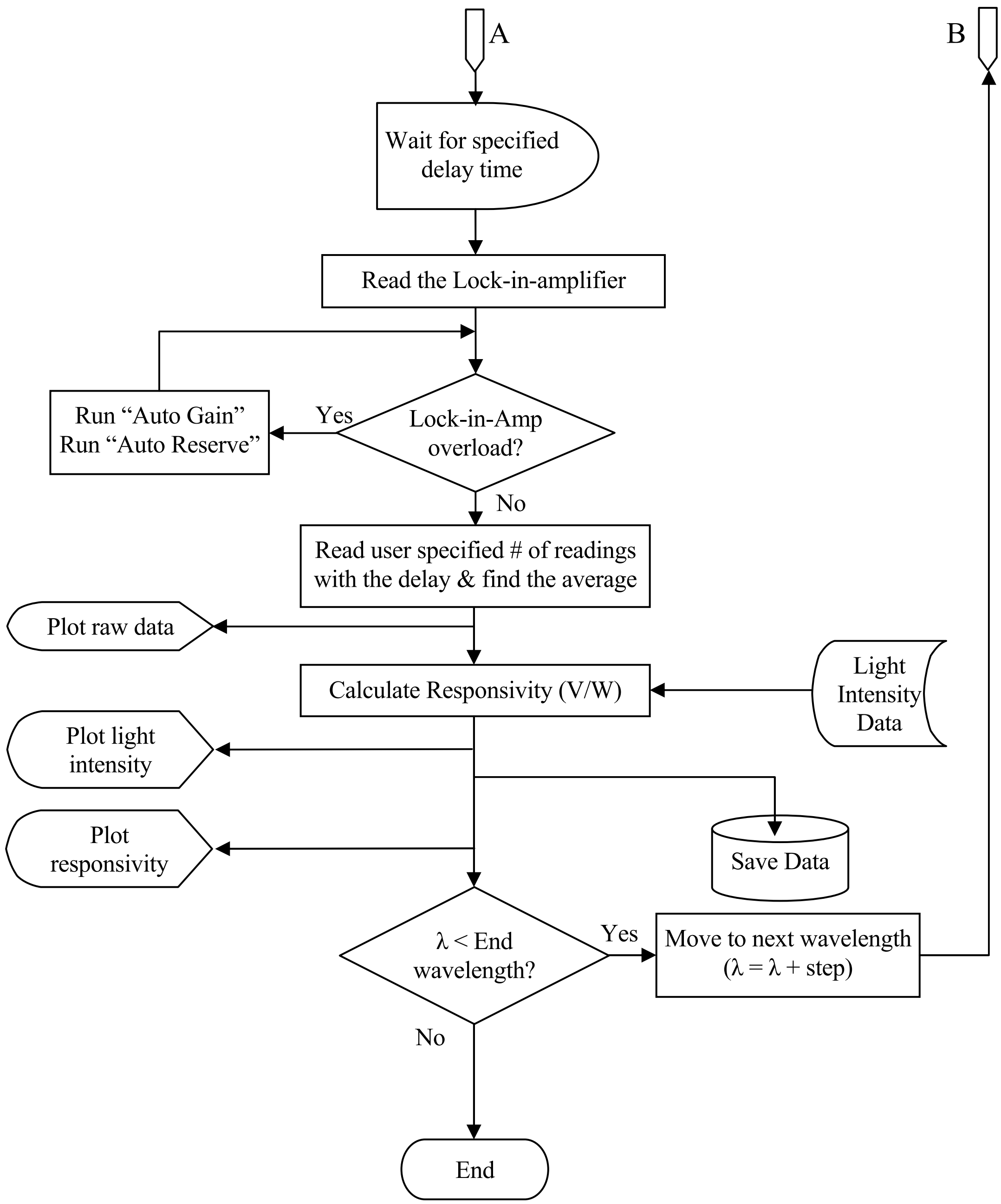


Figure B.20 The Lock-in-Amplifier control panel allows the user to change the lock-in amplifier settings and display different components of the output value such as the real part (x), imaginary part (y), phase angle (θ), and magnitude of the output (R). As an optional feature, the user can change the output voltage (-10 V to 10V) of the three auxiliary ports located in the back panel of the lock-in-amplifier.

B.2.3 Flow chart for the spectral responsivity (UV-VIS-IR) measurement (Spectral Products monochromator)




A
B
Wait for specified delay time
Read the Lock-in-amplifier
Run “Auto Gain”
Run “Auto Reserve”
Yes
Lock-in-Amp overload?
No
Read user specified # of readings with the delay & find the average
Plot raw data
Calculate Responsivity (V/W)
Light Intensity Data
Plot light intensity
Plot responsivity
Save Data
λ < End wavelength?
Yes
Move to next wavelength (λ = λ + step)
No
End

B.2.4 Source code

****

Module1.bas

****

```
'Copyrights - Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State
University

Public fMainForm As frmMain
Public MGPIB As Integer
Global Intensity(200 To 2960) As Double

Dim FMPos As Integer 'Current position of the filter wheel motor
Dim GMPos As Integer 'Current position of the grating motor
Dim D(1 To 8) As Integer 'To store 4 8 6 9 forwerd'
Dim CNF As Integer 'current data number send to filter wheel motor
Dim CNG As Integer 'current data number send to grating motor
Dim LPVal As Integer 'Last Port Value The last number sent to printer port

Sub Main()
    frmSplash.Show
    frmSplash.Refresh
    Set fMainForm = New frmMain
    Load fMainForm
    Unload frmSplash
    fMainForm.Show
    'd = GetSetting(App.Title, "Settings", "SpeedConst", 100000)
    Initialize.Visible = True
    DoEvents
    Unload Initialize

    fn% = FreeFile()
    Open App.Path & "\Intensity.dat" For Input As #fn%
    Input #fn%, a, b
    k = 200
    While Not EOF(fn%) And k <= 2960
        Input #fn%, a, Intensity(k)
        If a <> k Then
            MsgBox "Intensity data file has an error"
        End If
        k = k + 1
    Wend
    Close #fn%

End Sub
Sub Wait(Time_in_ms As Integer)
    st = Timer
    While Timer - st < Time_in_ms / 1000
        DoEvents
    Wend
End Sub

Function ChangeUnit(number As Single) As String
    If Abs(number) < 1 Then
        number = number * 1000
```

```
        unit = " mV"
    End If
    If Abs(number) < 1 Then
        number = number * 1000
        unit = " uV"
    End If
    If Abs(number) < 1 Then
        number = number * 1000
        unit = " nV"
    End If
    If Abs(number) < 1 Then
        number = number * 1000
        unit = " pV"
    End If
    ChangeUnit = Format(number, "000.0000") & unit
End Function

Sub MsgWait(Message As String)
    If Message = "" Then
        frmMsgBox.Visible = False
    Else
        With frmMsgBox
            .lblMessage.Caption = Message
            .Visible = True
        End With
    End If
End Sub

************************************************************************
DK480_GPIB.bas
************************************************************************
'Copyrights - Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State
University

Attribute VB_Name = "DK480_GPIB"
Global gblLightSelted As Integer  ' Selected light source 1= UV 2= VIS-NIR
Global gblDetectorSelted As Integer ' Selected Detector 1=UV enhansed Si
2=InGaAs  3=custom 1 etc.

Sub GoToLamdaCG(WLength&)
    'This will rotate the current grating to a new wavelength. "CG" means
current Grating.
    Dim Status%
    Status% = DKGoto%(WLength& * 100, fMainForm.comComm1, 100)
    If Status% < 128 Then
      gblCurWL& = WLength& * 100
    Else
      MsgBox "An error has occurred !", MB_ICONSTOP
    End If
End Sub

Sub GoToLamdaBG(WLength&)
    'This will select the best grating and the filter to goto new wavelength.
"BG" means Best Grating.
    Dim CWL&
    CWL& = gblCurWL& / 100
```

```
'********************** To select correct grating *******************
    Select Case WLength&

    Case Val(frmAdvSetup.txtWL(0)) To Val(frmAdvSetup.txtWL(1))
        If gblNGrtSelted& <> 1 Then
            GrtSel 1
        End If

    Case Val(frmAdvSetup.txtWL(1)) To Val(frmAdvSetup.txtWL(2))
        If gblNGrtSelted& <> 2 Then
            GrtSel 2
        End If

    Case Val(frmAdvSetup.txtWL(2)) To Val(frmAdvSetup.txtWL(3))
        If gblNGrtSelted& <> 3 Then
            GrtSel 3
        End If
    Case Else
        MsgBox "Wavelength " & WLength& & " nm is out of range. Please Check
the Advanced Settings."

    End Select

'********************** To select correct Filter *******************

    Select Case WLength&

    Case Val(frmAdvSetup.txtF(1)) To Val(frmAdvSetup.txtF(2))
        If gblFilter% <> 1 Then
            FltSel 1
        End If

    Case Val(frmAdvSetup.txtF(2)) To Val(frmAdvSetup.txtF(3))
        If gblFilter% <> 2 Then
            FltSel 2
        End If

    Case Val(frmAdvSetup.txtF(3)) To Val(frmAdvSetup.txtF(4))
        If gblFilter% <> 3 Then
            FltSel 3
        End If

    Case Val(frmAdvSetup.txtF(4)) To Val(frmAdvSetup.txtF(5))
        If gblFilter% <> 4 Then
            FltSel 4
        End If

    Case Val(frmAdvSetup.txtF(5)) To Val(frmAdvSetup.txtF(6))
        If gblFilter% <> 5 Then
            FltSel 5
        End If

    Case Val(frmAdvSetup.txtF(6)) To 2 * Val(frmAdvSetup.txtF(6))
        If gblFilter% <> 6 Then
            FltSel 6
        End If
```

```
    Case Is > 2 * Val(frmAdvSetup.txtF(6))
        If gblFilter% <> 6 Then
            FltSel 6
        End If
        MsgWait "!!!WARNING!!! You Don't have correct filter to block the
second orders after the " & Str(2 * Val(frmAdvSetup.txtF(6))) & " nm."
        DoEvents

    Case Else
        MsgBox "Wavelength " & WLength& & " nm is out of range. Please Check
the Advanced Settings."

    End Select

'*********************** To Select correct Light Source ***************
    Select Case WLength&

    Case Val(frmAdvSetup.txtLWL(0)) To Val(frmAdvSetup.txtLWL(1))
        If gblLightSelted <> 1 Then
            LightSel 1
        End If

    Case Val(frmAdvSetup.txtLWL(1)) To Val(frmAdvSetup.txtLWL(2))
        If gblLightSelted <> 2 Then
            LightSel 2
        End If

    Case Val(frmAdvSetup.txtLWL(2)) To Val(frmAdvSetup.txtLWL(3))
        If gblLightSelted <> 3 Then
            LightSel 3
        End If

    Case Else
        MsgBox "Wavelength " & WLength& & " nm is out of the range. System may
not have correct lamp. Please Check the Advanced Settings."

    End Select


'*********************** To select correct detector ******************
    If fMainForm.optMode(2).Value = True And fMainForm.optScan(1).Value = True
Then
        If gblDetectorSelted <> 3 Then
            DetSel 3
        End If
    Else
        Select Case WLength&
            Case Val(frmAdvSetup.txtDWL(0)) To Val(frmAdvSetup.txtDWL(1))
                If gblDetectorSelted <> 1 Then
                    DetSel 1
                End If

            Case Val(frmAdvSetup.txtDWL(1)) To Val(frmAdvSetup.txtDWL(2))
                If gblDetectorSelted <> 2 Then
                    DetSel 2
                End If
```

```
            Case Else
                MsgBox "Wavelength " & WLength& & " nm is out of the range.
System may not have correct lamp. Please Check the Advanced Settings."

        End Select
    End If

'********************** Go to wavelength **********************

    Status% = DKGoto%(WLength& * 100, fMainForm.comComm1, 100)
    If Status% < 128 Then
      gblCurWL& = WLength& * 100
    Else
      MsgBox "An error has occurred !", MB_ICONSTOP
    End If

End Sub

Sub GrtSel(GrtNo As Integer)
    MsgWait "Please wait... Selecting Grating # " & GrtNo
    DoEvents
    Status% = DKGrtsel%(GrtNo, fMainForm.comComm1, 500)
    If Status% < 128 Then
      gblNGrtSelted& = GrtNo
    Else
      MsgBox "An error has occurred !", MB_ICONSTOP
    End If
    MsgWait ""
    fMainForm.UpdateMainScreen
End Sub

Sub FltSel(FltNo As Integer)
    Dim Status%
    On Error GoTo cmdCanOKErr
    MsgWait "Please wait... Changing to Filter # " & FltNo
    DoEvents
    Status% = DKFilter%(FltNo, fMainForm.comComm1, 5) '5: 5s Timeout%
    If Status% < 128 Then gblFilter% = FltNo
    gblOkCancel% = IDOK
    fMainForm.UpdateMainScreen
    Wait 3000
    MsgWait ""
    DoEvents
    Exit Sub
cmdCanOKErr:
  MsgBox "Error code is : " & Err, MB_ICONEXCLAMATION

End Sub

Sub SetSlt(SlitNo As Integer, SlitWidth As Integer)
    Dim Status%, SlitByte%
    On Error GoTo cmdCanOKErr

    If SlitNo = 1 Then SlitByte% = 31 Else SlitByte% = 32

    MsgWait "Please wait... Changing to the slit Size"
```

```
    DoEvents
    Status% = DKSltAdjCal%(SlitByte%, SlitWidth, fMainForm.comComm1, 500)
    If Status% > 127 Then GoTo cmdCanOKErr

    Select Case SlitNo
      Case 1
        gblS1% = SlitWidth  ' S1 adjustment.
      Case 2
        gblS2% = SlitWidth  ' S2 adjustment.
    End Select
    gblOkCancel% = IDOK
    fMainForm.UpdateMainScreen
    MsgWait ""
    Exit Sub

cmdCanOKErr:
  MsgBox "Error code is : " & Err, MB_ICONEXCLAMATION

End Sub

Sub LightSel(LightNo As Integer) ' 1=UV 2=VIS-NIR
    MsgWait "Please wait... Changing the light Source"
    DoEvents
    RunMM frmStpMotSet.txtLmpStp(LightNo - 1)
    gblLightSelted = LightNo
   ' MsgBox "Please veryfy the lamp " & Trim(Str(LightNo)) & " power ON"

    Wait 3000
    MsgWait ""
End Sub

Sub DetSel(DetectorNo As Integer)
    MsgWait "Please wait... Changing the Detector"
    DoEvents
    RunDM frmStpMotSet.txtDetStp(DetectorNo - 1)
    MsgWait ""
    gblDetectorSelted = DetectorNo
    MsgBox "Please connect the Detector " & Trim(Str(DetectorNo)) & " to the
pre-amp"
    Wait 2000
End Sub
```

```
*********************************************************************
PCI8255V3.bas
*********************************************************************
'Copyrights – Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State
University

Global ShutOpen As Boolean
Global MotPowON As Boolean
```

```
Dim hWD As Long                              'HANDLE
Dim pciSlot As WD_PCI_SLOT
Dim dwAction As Long
Global hET_PCI8255_V3 As ET_PCI8255_V3_HANDLE
Dim Opened As Boolean
Dim MMPos As Integer                         'Mirror Motor Current position
Dim DMPos As Integer                         'Detector Motor Current position

Const PIB = &H0                              'Reset & PIB Cycle
Const AUXC = &H2                             'Auxuary Direction
Const AUXD = &H3                             'Auxuary Data
Const PA1 = &HC0                             'Port-PA 8255#1

Const PB1 = &HC4                             'Port-PB 8255#1
Const PC1 = &HC8                             'Port-PC 8255#1
Const PCC1 = &HCC&                           'Port-Control 8255#1

Const PA2 = &HD0                             'Port-PA 8255#2
Const PB2 = &HD4                             'Port-PB 8255#2
Const PC2 = &HD8                             'Port-PC 8255#2
Const PCC2 = &HDC                            'Port Control 8255#2

Const PA3 = &HE0                             'Port-PA 8255#3
Const PB3 = &HE4                             'Port-PB 8255#3
Const PC3 = &HE8                             'Port-PC 8255#3
Const PCC3 = &HEC                            'Port Control 8255#3

Const BIT0_ON = &H1                          '0000 0001 OR XXXX XXXX = XXXX XXX1
Const BIT0_OFF = &HFE                           '1111 1110 AND XXXX XXXX = XXXX
XXX0

Const BIT1_ON = &H2                          '0000 0010 OR XXXX XXXX = XXXX XX1X
Const BIT1_OFF = &HFD                           '1111 1101 AND XXXX XXXX = XXXX
XX0X

Const BIT2_ON = &H4                          '0000 0100 OR XXXX XXXX = XXXX X1XX
Const BIT2_OFF = &HFB                           '1111 1011 AND XXXX XXXX = XXXX
X0XX

Const BIT3_ON = &H8                          '0000 1000 OR XXXX XXXX = XXXX 1XXX
Const BIT3_OFF = &HF7                           '1111 0111 AND XXXX XXXX = XXXX
0XXX

Const BIT4_ON = &H10                         '0001 0000 OR XXXX XXXX = XXX1 XXXX
Const BIT4_OFF = &HEF                           '1110 1111 AND XXXX XXXX = XXX0
XXXX

Const BIT5_ON = &H20                         '0010 0000 OR XXXX XXXX = XX1X XXXX
Const BIT5_OFF = &HDF                           '1101 1111 AND XXXX XXXX = XX0X
XXXX

Const BIT6_ON = &H40                         '0100 0000 OR XXXX XXXX = X1XX XXXX
Const BIT6_OFF = &HBF                           '1011 1111 AND XXXX XXXX = X0XX
XXXX

Const BIT7_ON = &H80                         '1000 0000 OR XXXX XXXX = 1XXX XXXX
```

```
Const BIT7_OFF = &H7F                          '0111 1111 AND XXXX XXXX = 0XXX
XXXX


Dim DFSW(1 To 4) As Byte            'Data for Full Step Wave step sequence
Dim DHS(1 To 8) As Byte             'Data for Half Step sequence
Dim DFSHT(1 To 4) As Byte           'Data for Full Step High Torque sequence

Sub initializePCI8255()
    Dim InOutData As Byte
    Dim ControlWord As Byte

    'Step data for Full Step Wave. This is "or" data, "and" data = &H5 + &HC0
    DFSW(1) = &H5
    DFSW(2) = &H12
    DFSW(3) = &H9
    DFSW(4) = &H22

    'Step data for Full Steps High Torque. This is "or" data, "and" data = &H5
+ &HC0
    DFSHT(1) = &H27
    DFSHT(2) = &H17
    DFSHT(3) = &H1B
    DFSHT(4) = &H2B

    'Step data for Half Steps. This is "or" data, "and" data = &H5 + &HC0
    DHS(1) = &H5
    DHS(2) = &H27
    DHS(3) = &H22
    DHS(4) = &H2B
    DHS(5) = &H9
    DHS(6) = &H1B
    DHS(7) = &H12
    DHS(8) = &H3F


    Opened = False
    If       Not       PCI_Get_WD_handle(hWD)       Then       Exit       Sub
'Make sure Driver is loaded
    WD_Close (hWD)
    If Not Opened Then
       Opened                                                               =
ET_PCI8255_V3_LocateAndOpenBoard(ET_PCI8255_V3_DEFAULT_VENDOR_ID,
ET_PCI8255_V3_DEFAULT_DEVICE_ID, hET_PCI8255_V3)
    End If
    If Opened Then
       InOutData           =           ET_PCI8255_V3_ReadByte(hET_PCI8255_V3,
CByte(ET_PCI8255_V3_AD_BAR0), PIB)         'Read PIB Reset Port
       InputData          =          InOutData          And          BIT0_OFF
'Bit0 = EXTRST# = "0" (Reset:RES#)
       InOutData          =          InOutData          Or          BIT5_ON
'Bit5:4 = 11 = PIB Cycle Slowest
       InOutData = InOutData Or BIT4_ON
       Call                          ET_PCI8255_V3_WriteByte(hET_PCI8255_V3,
CByte(ET_PCI8255_V3_AD_BAR0), PIB, InOutData)    'Active RES#
```

```
       InOutData            =            ET_PCI8255_V3_ReadByte(hET_PCI8255_V3,
CByte(ET_PCI8255_V3_AD_BAR0), AUXD)        'Read Aux Data Port
       InputData           =          InOutData          And          BIT0_OFF
'Bit0 = Aux0 = "0" (Enable CS)
       InOutData           =           InOutData           Or            BIT4_ON
'Bit4 = Aux4 = "1" (Relay OFF)
       Call                            ET_PCI8255_V3_WriteByte(hET_PCI8255_V3,
CByte(ET_PCI8255_V3_AD_BAR0), AUXD, InOutData)   'Active Chips Select & Relay

       InOutData            =            ET_PCI8255_V3_ReadByte(hET_PCI8255_V3,
CByte(ET_PCI8255_V3_AD_BAR0), AUXC)        'Read Aux Direction Port
       InOutData           =           InOutData           Or            BIT4_ON
'Aux4 = Output
       InOutData           =           InOutData           Or            BIT0_ON
'Aux0 = Output
       Call                            ET_PCI8255_V3_WriteByte(hET_PCI8255_V3,
CByte(ET_PCI8255_V3_AD_BAR0), AUXC, InOutData)   'Setup Aux Direction

    End If
    ControlWord = &H88
    Call  ET_PCI8255_V3_WriteByte(hET_PCI8255_V3,  CByte(ET_PCI8255_V3_AD_BAR0),
PCC1, ControlWord)    'Control 8255#1 = PC (high bits) is Input

End Sub

Sub InitializeMM()
    'initialize Mirror motor
    For k = 1 To 5
        'HalfStep PA1, "F"
        FullStepHT PA1, "R"
        Wait 50
        DoEvents
    Next k
    If ReadPC1(5) = False Then
        For k = 1 To 30
            'HalfStep PA1, "F"
            FullStepHT PA1, "R"
            Wait 50
            DoEvents
        Next k
    End If
    k = 1
    While ReadPC1(5) = True And k < 160
        'HalfStep PA1, "R"
        FullStepHT PA1, "F"
        k = k + 1
        Wait 50
        DoEvents
    Wend
    MMPos = 0
    RunMM frmStpMotSet.txtLmpStp(1)
    gblLightSelted = 2
End Sub
Sub InitializeDM()
    'initialize Detector motor
    If ReadPC1(4) = True Then
        For k = 1 To 40
```

```
            FullStepHT PB1, "F"
            Wait 1
            DoEvents
        Next k
    End If
    k = 1
    While ReadPC1(4) = False And k < 1840
        FullStepHT PB1, "R"
        k = k + 1
        Wait 1
        DoEvents
    Wend
    DMPos = 0
    RunDM frmStpMotSet.txtDetStp(0)
    gblDetectorSelted = 1
End Sub



Sub RunMM(NewPos As Integer) 'To rotate Mirror motor to new position
    While MMPos < NewPos
        'HalfStep PA1, "F"
        'FullStepHT PA1, "F"
        FullStepW PA1, "F"
        MMPos = MMPos + 1
        DoEvents
        Wait 50
    Wend
    While MMPos > NewPos
        'HalfStep PA1, "R"
        'FullStepHT PA1, "R"
        FullStepW PA1, "R"
        MMPos = MMPos - 1
        DoEvents
        Wait 50
    Wend

End Sub

Sub RunDM(NewPos As Integer) 'To rotate Detector motor to new position
    While DMPos < NewPos
        'HalfStep Pb1, "F"
        FullStepHT PB1, "F"
        'FullStepW PB1, "F"
        DMPos = DMPos + 1
        DoEvents
        Wait 1
    Wend
    While DMPos > NewPos
        'HalfStep Pb1, "R"
        FullStepHT PB1, "R"
        'FullStepW PB1, "R"
        DMPos = DMPos - 1
        DoEvents
        Wait 1
    Wend
```

```
End Sub


Sub FullStepW(MotorPort As Long, Direction_F_R As String) 'Run the motor 1 step
using Full Step Wave sequence
    Dim InOutData As Byte
    Dim CheckData As Byte
    Dim SNo As Integer
    SNo = 0

    InOutData              =              ET_PCI8255_V3_ReadByte(hET_PCI8255_V3,
CByte(ET_PCI8255_V3_AD_BAR0), MotorPort)       ' Read Output Latch Port

    For k = 1 To 4
        If (InOutData Or DFSW(k)) = InOutData Then SNo = k  ' find out current
step no
    Next k
    If SNo = 0 Then SNo = 1

    If UCase(Direction_F_R) = "F" Then
        If SNo < 4 Then
            SNo = SNo + 1
        Else
            SNo = 1
        End If
    Else
        If SNo > 1 Then
            SNo = SNo - 1
        Else
            SNo = 4
        End If
    End If
    InOutData = InOutData Or DFSW(SNo)              'make "1"s in the new bit
pattern witout changeing unused bits
    InOutData = InOutData And (DFSW(SNo) + &HC0) 'make "0"s in the new bit
pattern witout changeing unused bits

    Call  ET_PCI8255_V3_WriteByte(hET_PCI8255_V3,  CByte(ET_PCI8255_V3_AD_BAR0),
MotorPort, InOutData)   ' Update Port

End Sub

Sub FullStepHT(MotorPort As Long, Direction_F_R As String) 'Run the motor 1
step using Full Step High Torque sequence
    Dim InOutData As Byte
    Dim CheckData As Byte
    Dim SNo As Integer
    SNo = 0

    InOutData              =              ET_PCI8255_V3_ReadByte(hET_PCI8255_V3,
CByte(ET_PCI8255_V3_AD_BAR0), MotorPort)       ' Read Output Latch Port

    For k = 1 To 4
        If (InOutData Or DFSHT(k)) = InOutData Then SNo = k  ' find out current
step no
    Next k
    If SNo = 0 Then SNo = 1
```

```
    If UCase(Direction_F_R) = "F" Then
        If SNo < 4 Then
            SNo = SNo + 1
        Else
            SNo = 1
        End If
    Else
        If SNo > 1 Then
            SNo = SNo - 1
        Else
            SNo = 4
        End If
    End If
    InOutData = InOutData Or DFSHT(SNo)    'make "1"s in the new bit pattern
witout changeing unused bits
    InOutData = InOutData And (DFSHT(SNo) + &HC0) 'make "0"s in the new bit
pattern witout changeing unused bits

    Call ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, CByte(ET_PCI8255_V3_AD_BAR0),
MotorPort, InOutData)   ' Update Port

End Sub

Sub HalfStep(MotorPort As Long, Direction_F_R As String) 'Run the motor 1 step
using Half Step Wave sequence
    Dim InOutData As Byte
    Dim CheckData As Byte
    Dim SNo As Integer
    SNo = 0

    InOutData              =              ET_PCI8255_V3_ReadByte(hET_PCI8255_V3,
CByte(ET_PCI8255_V3_AD_BAR0), MotorPort)       ' Read Output Latch Port

    For k = 1 To 8
        If (InOutData Or DHS(k)) = InOutData Then SNo = k   ' To find out
current step no
    Next k
    If SNo = 0 Then SNo = 1

    If UCase(Direction_F_R) = "F" Then
        If SNo < 8 Then
            SNo = SNo + 1
        Else
            SNo = 1
        End If
    Else
        If SNo > 1 Then
            SNo = SNo - 1
        Else
            SNo = 8
        End If
    End If
    InOutData = InOutData Or DHS(SNo)              'make "1"s in the new bit
pattern witout changeing unused bits
    InOutData = InOutData And (DHS(SNo) + &HC0) 'make "0"s in the new bit
pattern witout changeing unused bits
```

```
    Call  ET_PCI8255_V3_WriteByte(hET_PCI8255_V3,  CByte(ET_PCI8255_V3_AD_BAR0),
MotorPort, InOutData)    ' Update Port

End Sub

Function ReadPC1(BitNo) As Boolean
    Dim InOutData As Byte
    Dim CheckData As Byte
    Dim BITx_ON As Byte
    BITx_ON = 2 ^ BitNo
    InOutData              =              ET_PCI8255_V3_ReadByte(hET_PCI8255_V3,
CByte(ET_PCI8255_V3_AD_BAR0), PC1)    'Read Input/Output Latch of Port-C1
    CheckData = InOutData And BITx_ON
    If CheckData = BITx_ON Then ReadPC1 = True Else ReadPC1 = False

End Function
Sub Shutter(open_close As String)
   Dim InOutData As Byte

   InOutData              =              ET_PCI8255_V3_ReadByte(hET_PCI8255_V3,
CByte(ET_PCI8255_V3_AD_BAR0), PA1)        ' Read Output Latch Port-A

   If UCase(open_close) = "CLOSE" Then
      InOutData           =           InOutData           And           BIT6_OFF
' PA6 = "0" (1011 1111)
      ShutOpen = False
   Else
      InOutData           =           InOutData           Or            BIT6_ON
' PA6 = "1" (0100 0000)
      ShutOpen = True
   End If

   Call  ET_PCI8255_V3_WriteByte(hET_PCI8255_V3,  CByte(ET_PCI8255_V3_AD_BAR0),
PA1, InOutData)   ' Update Port-A

End Sub

Sub MotorPower(On_Off As String)
   Dim InOutData As Byte

   InOutData              =              ET_PCI8255_V3_ReadByte(hET_PCI8255_V3,
CByte(ET_PCI8255_V3_AD_BAR0), PA1)        ' Read Output Latch Port-A

   If UCase(On_Off) = "ON" Then
      InOutData           =           InOutData           And           BIT7_OFF
' PA6 = "0" (1011 1111)
      MotPowON = True
   Else
      InOutData           =           InOutData           Or            BIT7_ON
' PA6 = "1" (0100 0000)
      MotPowON = False
   End If

   Call  ET_PCI8255_V3_WriteByte(hET_PCI8255_V3,  CByte(ET_PCI8255_V3_AD_BAR0),
PA1, InOutData)   ' Update Port-A

End Sub
```

```
Sub ClosePort()
    If Opened Then Call ET_PCI8255_V3_Close(hET_PCI8255_V3)
    Opened = False
End Sub
```

****
SR850_GPIB.bas
****

```
‘Copyrights – Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State University

Attribute VB_Name = "SR850_GPIB"
Global CStep As Integer 'Current step No of Stepper motor
Global IniOK As Boolean ' To check correctly Initialized the stepper motor
Global PauseStop As Integer 'To check user press [Pause] or [Stop] play=0 pause=1 Stop =2
Global AutoTrack As Boolean
Const BDINDEX = 0                    ' Board Index
Const PRIMARY_ADDR_OF_DMM = 8        ' Primary address of device
Const NO_SECONDARY_ADDR = 0          ' Secondary address of device
Const Timeout = T10s                 ' Timeout value = 10 seconds
Const EOTMODE = 1                    ' Enable the END message
Const EOSMODE = 0                    ' Disable the EOS mode

Const ARRAYSIZE = 1024               ' Size of read buffer

Dim ErrMsg As String * 100
Dim Dev As Integer
Dim ErrorMnemonic
Dim ValueStr As String * ARRAYSIZE
Dim Response As Integer

'To read Data (R)from SR850 (while scaning)
Function SR850Data()
    ilwrt Dev%, "OUTP? 3", 7
    If (ibsta And EERR) Then
        Call GPIBCleanup("Unable to write to device")
    End If

    ilrd Dev%, ValueStr, Len(ValueStr)
    If (ibsta And EERR) Then
        Call GPIBCleanup("Unable to read from device")
    End If
    SR850Data = Val(Left$(ValueStr, ibcntl - 1))

End Function
'To read averaged Data from SR850
Function SR850AvgData(NoOfAvg As Integer, Delay As Integer)
    DataSum = 0
    For k = 1 To NoOfAvg
        DataSum = DataSum + SR850Data
        Wait Delay
    Next k
```

```
        SR850AvgData = DataSum / NoOfAvg
End Function

'To write command to SR850
Sub WriteToSR850(CommandString As String)
    ilwrt Dev%, CommandString, Len(CommandString)
    If (ibsta And EERR) Then
        Call GPIBCleanup("Unable to write to device")
    End If
End Sub

'To read Data from SR850
Function ReadSR850()
    ilrd Dev%, ValueStr, Len(ValueStr)
    If (ibsta And EERR) Then
        Call GPIBCleanup("Unable to read from device")
    End If
    ReadSR850 = Val(Left$(ValueStr, ibcntl - 1))
End Function

Sub InitializeGPIB()
    Dev% = ildev(BDINDEX, PRIMARY_ADDR_OF_DMM, NO_SECONDARY_ADDR, _
                 Timeout, EOTMODE, EOSMODE)
    If (ibsta And EERR) Then
        ErrMsg = "Unable to open device" & Chr(13) & "ibsta = &H" & _
        Chr(13) & Hex(ibsta) & "iberr = " & iberr
        MsgBox ErrMsg, vbCritical, "Error"
        End
    End If
    ' The application resets the GPIB portion of the device by calling
    ' ilclr.
    ilclr Dev%
    If (ibsta And EERR) Then
        Call GPIBCleanup("Unable to clear device")
    End If
    ilwrt Dev%, "OUTX 1", 6
    If (ibsta And EERR) Then
        Call GPIBCleanup("Unable to write to device")
    End If

End Sub

Sub GPIBCleanup(Msg$)

    ' After each GPIB call, the application checks whether the call
    ' succeeded. If an NI-488.2 call fails, the GPIB driver sets the
    ' corresponding bit in the global status variable. If the call
    ' failed, this procedure prints an error message, takes the device
    ' offline and exits.

    ErrorMnemonic = Array("EDVR", "ECIC", "ENOL", "EADR", "EARG", _
                          "ESAC", "EABO", "ENEB", "EDMA", "", _
                          "EOIP", "ECAP", "EFSO", "", "EBUS", _
                          "ESTB", "ESRQ", "", "", "", "ETAB")

    ErrMsg$ = Msg$ & Chr(13) & "ibsta = &H" & Hex(ibsta) & Chr(13) _
            & "iberr = " & iberr & " <" & ErrorMnemonic(iberr) & ">"
```

```
    MsgBox ErrMsg$, vbCritical, "Error"
    ilonl Dev%, 0

End Sub

Sub InitializeSR850()
    WriteToSR850 "FMOD2" 'reference source "external"
    WriteToSR850 "RSLP1" 'reference slope sine zero crossing (i=0), TTL rising
edge (i=1), TTL falling edge (i=2)
    WriteToSR850 "HARM1" 'detection harmonic 1
    WriteToSR850 "ISRC0"
    WriteToSR850 "IGND0"
    WriteToSR850 "ICPL0"  'input coupling AC
    WriteToSR850 "ILIN0"   'input line notch filter status no filters (i=0),
Line notch in (i=1), 2xLine notch in (i=2) or Both notch filters in (i=3)..
    WriteToSR850 "OFSL1"
    WriteToSR850 "SYNC1"
    WriteToSR850 "SMOD1" 'The SMOD command sets or queries the screen format.
The parameter i selects Single or full screen display (i=0), or Up/Down dual
display (i=1).
    WriteToSR850 "MNTR1" 'monitor display mode.
    WriteToSR850 "DTYP1,2" 'display type.
    WriteToSR850 "DTYP2,2"
    WriteToSR850 "DTRC1,3 "  'displayed trace number.
    WriteToSR850 "DTRC2,4 "  'displayed trace number.
    'WriteToSR850 ""
    'WriteToSR850 ""

End Sub

Function TimeConstant() As String
    WriteToSR850 "OFLT?"
    TCNo = ReadSR850
    Select Case TCNo
        Case 0
            TimeConstant = "10 us"
        Case 1
            TimeConstant = "30 us"
        Case 2
            TimeConstant = "100 us"
        Case 3
            TimeConstant = "300 us"
        Case 4
            TimeConstant = "1 ms"
        Case 5
            TimeConstant = "3 ms"
        Case 6
            TimeConstant = "10 ms"
        Case 7
            TimeConstant = "30 ms"
        Case 8
            TimeConstant = "100 ms"
        Case 9
            TimeConstant = "300 ms"
        Case 10
            TimeConstant = "1 s"
        Case 11
```

```
            TimeConstant = "3 s"
        Case 12
            TimeConstant = "10 s"
        Case 13
            TimeConstant = "30 s"
        Case 14
            TimeConstant = "100 s"
        Case 15
            TimeConstant = "300 s"
        Case 16
            TimeConstant = "1 ks"
        Case 17
            TimeConstant = "3 ks"
        Case 18
            TimeConstant = "10 ks"
        Case 19
            TimeConstant = "30 ks"
    End Select
End Function

Function TimeConsSec() As Single
    WriteToSR850 "OFLT?"
    TCNo = ReadSR850
    Select Case TCNo
        Case 0
            TimeConsSec = 0.00001
        Case 1
            TimeConsSec = 0.00003
        Case 2
            TimeConsSec = 0.0001
        Case 3
            TimeConsSec = 0.0003
        Case 4
            TimeConsSec = 0.001
        Case 5
            TimeConsSec = 0.003
        Case 6
            TimeConsSec = 0.01
        Case 7
            TimeConsSec = 0.03
        Case 8
            TimeConsSec = 0.1
        Case 9
            TimeConsSec = 0.3
        Case 10
            TimeConsSec = 1
        Case 11
            TimeConsSec = 3
        Case 12
            TimeConsSec = 10
        Case 13
            TimeConsSec = 30
        Case 14
            TimeConsSec = 100
        Case 15
            TimeConsSec = 300
        Case 16
```

```
            TimeConsSec = 1000
        Case 17
            TimeConsSec = 3000
        Case 18
            TimeConsSec = 10000
        Case 19
            TimeConsSec = "30 ks"
    End Select
End Function
Function Sensitivity() As Double
    WriteToSR850 "SENS?"
    SensNo = ReadSR850
    Select Case SensNo
        Case 0
            Sensitivity = 0.000000002
        Case 1
            Sensitivity = 0.000000005
        Case 2
            Sensitivity = 0.00000001
        Case 3
            Sensitivity = 0.00000002
        Case 4
            Sensitivity = 0.00000005
        Case 5
            Sensitivity = 0.0000001
        Case 6
            Sensitivity = 0.0000002
        Case 7
            Sensitivity = 0.0000005
        Case 8
            Sensitivity = 0.000001
        Case 9
            Sensitivity = 0.000002
        Case 10
            Sensitivity = 0.000005
        Case 11
            Sensitivity = 0.00001
        Case 12
            Sensitivity = 0.00002
        Case 13
            Sensitivity = 0.00005
        Case 14
            Sensitivity = 0.0001
        Case 15
            Sensitivity = 0.0002
        Case 16
            Sensitivity = 0.0005
        Case 17
            Sensitivity = 0.001
        Case 18
            Sensitivity = 0.002
        Case 19
            Sensitivity = 0.005
        Case 20
            Sensitivity = 0.01
        Case 21
            Sensitivity = 0.02
```

```
        Case 22
            Sensitivity = 0.05
        Case 23
            Sensitivity = 0.1
        Case 24
            Sensitivity = 0.2
        Case 25
            Sensitivity = 0.5
        Case 26
            Sensitivity = 1
    End Select
End Function

Function ChkOverload(Point As Integer) As Boolean
    'Point 0 = INPUT or RESRV overload
    'Point 1 = FILTR overload
    'Point 2 = OUTPT overload

    WriteToSR850 "LIAS?" & Trim(Str(Point))
    ChkOverload = ReadSR850

End Function

Sub SetGain() 'if overload detected decrease the sensitivity
    WriteToSR850 "SENS?"
    cgain = ReadSR850
    If ChkOverload(2) Then
        MsgWait "Output Overload Detected! Gain is increasing...."
        Wait 1000

        While ChkOverload(2) And cgain < 26
            WriteToSR850 "SENS?"
            cgain = ReadSR850
            WriteToSR850 "SENS" & Trim(Str(cgain + 1))
            Wait 2000
        Wend
        MsgWait ""
        If cgain = 26 Then
            MsgBox "Output is too high ! Please reduse pre amp gain or light
intensity"
        End If

    ElseIf AutoTrack = True Then
        If SR850AvgData(5, CInt(TimeConsSec * 1000)) < 0.1 * Sensitivity Then
            WriteToSR850 "SENS" & Trim(Str(cgain - 1))
            Wait 2000
        End If
    End If

End Sub


***********************************************************************
ModSaveFormData.bas
***********************************************************************
```

```
'Copyrights - Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State University

Attribute VB_Name = "ModSaveFormData"
Option Explicit


Public Sub ReadFormData(frm As Form, FileName As String, Optional ByRef LoadError As Boolean)

Dim ctrl As Control, f As Integer, cval As Variant, cname As String, idx As Variant, cidx As Variant
Dim j As Integer, lstcnt As Integer, lstidx As Integer

f = FreeFile()
On Error Resume Next
Open FileName For Input As #f
If Err.Number = 53 Then Exit Sub 'File not found
Err.Clear

Input #f, cname
If Not cname Like App.FileDescription Then
   Close #f
   MsgBox "File is not a " & App.FileDescription & " Data File", vbExclamation + vbOKOnly, "Error reading file"
   LoadError = True
   Exit Sub
End If

While Not EOF(f)
    Input #f, cname, cval, idx
    For Each ctrl In frm.Controls
        Err.Clear
        cidx = Null
        With ctrl
            If .Name = cname Then
               cidx = .Index
                  If TypeName(cidx) = "Null" Or cidx = idx Then
                      If TypeOf ctrl Is CheckBox Then
                          .value = cval

                      ElseIf TypeOf ctrl Is OptionButton Then
                          .value = cval

                      ElseIf TypeOf ctrl Is TextBox Then
                          .Text = cval

                      ElseIf TypeOf ctrl Is ComboBox Then
                          .ListIndex = cval

                      ElseIf TypeOf ctrl Is ListBox Then
                           lstidx = cval
                           Input #f, lstcnt
                           .Clear
                           For j = 1 To lstcnt
                              Input #f, cval
```

```
                              .AddItem cval
                           Next
                           .ListIndex = lstidx
                      End If
                Exit For
                End If
            End If
        End With
    Next ctrl
Wend

Close #f

End Sub


Public Sub SaveFormData(frm As Form, FileName As String)

Dim ctrl As Control, f As Integer, idx As Variant, j As Integer
f = FreeFile()

Open FileName For Output As #f

Write #f, App.FileDescription
On Error Resume Next

For Each ctrl In frm.Controls
    Err.Clear
    idx = Null
    With ctrl
        idx = .Index
        If TypeOf ctrl Is CheckBox Then
            Print #f, .Name
            Write #f, .value
            Write #f, idx

         ElseIf TypeOf ctrl Is OptionButton Then
            Print #f, .Name
            Write #f, .value
            Write #f, idx

        ElseIf TypeOf ctrl Is TextBox Then
            Print #f, .Name
            Write #f, .Text
            Write #f, idx

        ElseIf TypeOf ctrl Is ComboBox Then
            Print #f, .Name
            Write #f, .ListIndex
            Write #f, idx

        ElseIf TypeOf ctrl Is ListBox Then
            Print #f, .Name
            Write #f, .ListIndex
            Write #f, idx
            Write #f, .ListCount
            For j = 1 To .ListCount
```

```
            Write #f, .List(j - 1)
         Next


      End If

   End With
Next ctrl

Close #f

End Sub
```

****
frmInitialize.frm
****

```
'Copyrights – Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State
University

Private Sub cmdIgnore_Click()
    IniOK = False
    Me.Visible = False
End Sub

Private Sub Form_Activate()

  Dim I%, Baud$
 ' On Error GoTo FormErr

  IniOK = False
  pBarInitialize.Value = 10

  gblIniExist% = GetDkInit%()
  If gblIniExist% Then
    If COMInit%(gblPortNo%, gblBaud%) Then
      'Initializing dk successfully. Do nothing.
    Else
      SetCOM.Show MODAL
    End If
  Else
    SetCOM.Show MODAL
  End If
  If gblInitCom% Then
    'If CM110, gblType% = 0; if CM112, gblType%=1 or 254
    If gblType% = 0 Then    'if CM110
      'mnuType.Enabled = False
      'mnuIncDec2.Enabled = False
    Else
      'mnuType.Enabled = True
      'mnuIncDec2.Enabled = True
    End If
    gblFilter% = 1
    InitQueries
    fMainForm.UpdateMainScreen
  End If
```

```
  'AlignGotoBar

pBarInitialize.Value = 60
lblMessage = "Initialize Lock-in Amplifier SR850"

InitializeGPIB
pBarInitialize.Value = 80

InitializeSR850
pBarInitialize.Value = 100
fMainForm.UpdateMainScreen

pBarInitialize.Value = 5
lblMessage = "Initializing PCI8255 interface card"
DoEvents
initializePCI8255
pBarInitialize.Value = 20

lblMessage = "Power up Motor Powersupply"
MotorPower "ON"
Wait 1000
Shutter "Open"
fMainForm.UpdateMainScreen
pBarInitialize.Value = 30

lblMessage = "Checking Shutter..."
DoEvents
Shutter "Close"
Wait 500
Shutter "Open"

lblMessage = "Initializing Mirror Stepper Motor"
DoEvents
InitializeMM
pBarInitialize.Value = 50

lblMessage = "Initializing Detector Stepper Motor"
DoEvents
InitializeDM
pBarInitialize.Value = 100

Wait 500

IniOK = True
Me.Visible = False

FormResume:
  Exit Sub

FormErr:
  MsgBox "Error code is : " & Err, MB_ICONEXCLAMATION
  Resume FormResume

End Sub

Private Sub AlignGotoBar()
  Dim LeftVar!, TopVar!, HeightVar!, WidthVar!
```

```
  Dim I%

 ' 'HeightVar! = picStep(0).Height
 ' TopVar! = picStep(0).Top

  'For I% = 1 To 4
  '  WidthVar! = picStep(I%).Width
  '  LeftVar! = picStep(I% - 1).Left + picStep(I% - 1).Width
   ' picStep(I%).Move LeftVar!, TopVar!, WidthVar!, HeightVar!
 ' Next I%
End Sub
```

****

frmMain.frm

****

```
'Copyrights - Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State University

Private Bpress As Boolean
Private Declare Function SendMessage Lib "user32" Alias "SendMessageA" (ByVal hwnd As Long, ByVal wMsg As Long, ByVal wParam As Long, ByVal lParam As Any) As Long
Const EM_UNDO = &HC7
Private Declare Function OSWinHelp% Lib "user32" Alias "WinHelpA" (ByVal hwnd&, ByVal HelpFile$, ByVal wCommand%, dwData As Any)

Private Sub cmdCyclePlay_Click()
    If IniOK = False Then
        ch = MsgBox("Hardware NOT initialized." & Chr(10) & Chr(13) & "Would you like to start initialize now", vbYesNo, "Initialize Error")
        If ch = 7 Then
            Exit Sub
        End If
        Initialize.Visible = True
        Unload Initialized
    End If
    PauseStop = 0
    If txtStart < 400 Then txtStart = 400
    If txtStart > 1250 Then txtStart = 1250
    If Abs(gblCurWL& - txtStart.Text) < Abs(gblCurWL& - txtStop.Text) Then
        ActiveForm.PlayScan txtStart.Text, txtStep.Text, txtStop.Text, txtDelay, txtAvg, False
        ActiveForm.PlayScan txtStop.Text, -txtStep.Text, txtStart.Text, txtDelay, txtAvg, False
    Else
        ActiveForm.PlayScan txtStop.Text, -txtStep.Text, txtStart.Text, txtDelay, txtAvg, False
        ActiveForm.PlayScan txtStart.Text, txtStep.Text, txtStop.Text, txtDelay, txtAvg, False
    End If

End Sub

Private Sub cmdFF_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
    If Button = 2 Then
    '     GoToLamdaCG InputBox("Enter Wavelength that You need to go", "Go to
Lamda", 1000)
    Else
        Bpress = True
        While Bpress = True
         '   GoG 1, "F"
         '   Wait 0.3
           ' DoEvents
        Wend
    End If
End Sub

Private Sub cmdFF_MouseUp(Button As Integer, Shift As Integer, X As Single, Y
As Single)
    Bpress = False
End Sub

Private Sub cmdFFU_MouseUp(Button As Integer, Shift As Integer, X As Single, Y
As Single)
    WriteToSR850 "OUTP?3"
    UpdateDispData ReadSR850
    Bpress = False
End Sub

Private Sub cmdMotPow_Click()
    If MotPowON Then
        MotorPower "OFF"
        shpMPwLED.FillColor = &H0
    Else
        MotorPower "ON"
        shpMPwLED.FillColor = &HFF&
        MsgWait "Please wait Initilizing Stepper Motors"
        InitializeMM
        InitializeDM
        MsgWait ""
    End If
End Sub

Private Sub cmdPause_Click()
    If PauseStop = 1 Then
        MsgWait ""
        PauseStop = 0
    Else
        MsgWait "Scan Pause by User. Press Pause agian to continue."
        DoEvents
        PauseStop = 1
        Me.cmdPlay.Enabled = False

    End If
End Sub

Private Sub cmdREV_MouseDown(Button As Integer, Shift As Integer, X As Single,
Y As Single)
    If Button = 2 Then
      ' GotoLamdacg InputBox("Enter Wavelength that You need to go", "Go to
Lamda", 1000)
```

```
    Else
        Bpress = True
        While Bpress = True
           ' GotoLamda gblCurWL& - 1
            'DoEvents
        Wend
    End If
End Sub

Private Sub cmdREV_MouseUp(Button As Integer, Shift As Integer, X As Single, Y
As Single)
    WriteToSR850 "OUTP?3"
    UpdateDispData ReadSR850
    Bpress = False
End Sub

Private Sub cmdREVU_MouseUp(Button As Integer, Shift As Integer, X As Single, Y
As Single)
    WriteToSR850 "OUTP?3"
    UpdateDispData ReadSR850
    Bpress = False
End Sub

Private Sub cmdPlay_Click()
    cmdPlay.Enabled = False
    cmdPause.Enabled = True
    cmdStop.Enabled = True
    PauseStop = 0

    If optMode(2).Value = True And optScan(1).Value = True Then
        If IniOK = False Then
            ch  =  MsgBox("Hardware  NOT  initialized."  &  Chr(10)  &  Chr(13)  &
"Would you like to start initialize now", vbYesNo, "Initialize Error")
            If ch = 7 Then
                Exit Sub
            End If
            Initialize.Visible = True
            Unload Initialized
        End If
        'If txtStart < 400 Then txtStart = 400
        'If txtStart > 1250 Then txtStart = 1250
        'ActiveForm.SetScale txtStart, txtStop
        ActiveForm.ScanSample   txtStart.Text,   txtStep.Text,   txtStop.Text,
txtDelay, txtAvg
    End If

    If optMode(2).Value = True And optScan(0).Value = True Then
        If IniOK = False Then
            ch  =  MsgBox("Hardware  NOT  initialized."  &  Chr(10)  &  Chr(13)  &
"Would you like to start initialize now", vbYesNo, "Initialize Error")
            If ch = 7 Then
                Exit Sub
            End If
            Initialize.Visible = True
            Unload Initialized
        End If
```

```
        ActiveForm.Calibrate    txtStart.Text,    txtStep.Text,    txtStop.Text,
txtDelay, txtAvg
    End If

    'ABS Background mode
    If optMode(1).Value = True And optScan(0).Value = True Then
         ActiveForm.ScanABSBakGnd  txtStart.Text,  txtStep.Text,  txtStop.Text,
txtDelay, txtAvg
    End If



End Sub

Private Sub cmdStop_Click()
    PauseStop = 2
    cmdPlay.Enabled = True
    cmdPause.Enabled = False
    cmdStop.Enabled = False
    MsgWait ""
End Sub

Private Sub frmShutter_Click()
    If shpShutter.FillColor = &H8000000F Then
        Shutter "close"
        lblShutter.Caption = "Closed"
        shpShutter.FillColor = &H0
    Else
        Shutter "open"
        lblShutter.Caption = "Open"
        shpShutter.FillColor = &H8000000F
    End If
End Sub

Private Sub MDIForm_Load()
    ReadFormData Me, App.Path & "\Settings_Main.dat"

    Me.Left = GetSetting(App.Title, "Settings", "MainLeft", 1000)
    Me.Top = GetSetting(App.Title, "Settings", "MainTop", 1000)
    Me.Width = GetSetting(App.Title, "Settings", "MainWidth", 7500)
    Me.Height = GetSetting(App.Title, "Settings", "MainHeight", 6500)
    MGPIB = GetSetting(App.Title, "Settings", "GPIB", 8)
    'Me.txtGPIB = MGPIB
    LoadNewDoc
End Sub

Private Sub LoadNewDoc()
    Static lDocumentCount As Long
    Dim frmD As frmDocument
    lDocumentCount = lDocumentCount + 1
    Set frmD = New frmDocument
    frmD.Caption = "Spectrum Book " & lDocumentCount
    frmD.Width = Me.Width - 5960
    frmD.Height = Me.Height - 2100
    frmD.Show
    frmD.Refresh
End Sub
```

```
Private Sub MDIForm_Unload(Cancel As Integer)
    SaveFormData Me, App.Path & "\Settings_Main.dat"

    If Me.WindowState <> vbMinimized Then
        SaveSetting App.Title, "Settings", "MainLeft", Me.Left
        SaveSetting App.Title, "Settings", "MainTop", Me.Top
        SaveSetting App.Title, "Settings", "MainWidth", Me.Width
        SaveSetting App.Title, "Settings", "MainHeight", Me.Height
        SaveSetting App.Title, "Settings", "GPIB", MGPIB
    End If
    MotorPower "OFF"
    Shutter "Close"
    ilonl Dev%, 0
    ClosePort
    Close All
    End
End Sub

Private Sub optMode_Click(Index As Integer)
    If Index = 0 Then
        frmMonitor.Show vbModal, Me
    End If
    If Index = 1 Then
       optScan(0).Caption = "BakGnd"
    End If
    If Index = 2 Then
        optScan(0).Caption = "Calibrate"
    End If

End Sub

Private Sub tbAdvStp_Click()
    frmAdvSetup.Visible = True
End Sub

Private Sub tbChopper_Click()
 frmChopper.Show MODAL
End Sub

Private Sub tbFilter_Click()
  If gblInitCom% Then
    ctrFilter.Show MODELESS
    UpdateMainScreen
  Else
    MsgBox "Did you initialize a serial port ? "
  End If
End Sub


Private Sub tbSR850_Click()
    frmSR850.Visible = True
End Sub


Private Sub tbToolBar_ButtonClick(ByVal Button As MSComctlLib.Button)
    On Error Resume Next
```

```
    Select Case Button.Key
        Case "New"
            LoadNewDoc
        Case "Open"
            mnuFileOpen_Click
        Case "Save"
            mnuFileSave_Click
        Case "Multi"
            Static v
            If v = 0 Then v = 1 Else v = 0
            frmSR850.Visible = v
            Button.Value = v
        Case "IV"

        Case "Setting"

        Case "Print"
            mnuFilePrint_Click
        Case "Cut"
            mnuEditCut_Click
        Case "Copy"
            mnuEditCopy_Click
        Case "Paste"
            mnuEditPaste_Click
        Case "Bold"
          '  ActiveForm.rtfText.SelBold = Not ActiveForm.rtfText.SelBold
          '     Button.Value   =   IIf(ActiveForm.rtfText.SelBold,   tbrPressed,
tbrUnpressed)
        Case "Italic"
           ' ActiveForm.rtfText.SelItalic = Not ActiveForm.rtfText.SelItalic
          '     Button.Value   =   IIf(ActiveForm.rtfText.SelItalic,   tbrPressed,
tbrUnpressed)
        Case "Underline"
        '                        ActiveForm.rtfText.SelUnderline      =      Not
ActiveForm.rtfText.SelUnderline
         '     Button.Value  =  IIf(ActiveForm.rtfText.SelUnderline,  tbrPressed,
tbrUnpressed)
        Case "Align Left"
         '   ActiveForm.rtfText.SelAlignment = rtfLeft
        Case "Center"
         '   ActiveForm.rtfText.SelAlignment = rtfCenter
        Case "Align Right"
         '   ActiveForm.rtfText.SelAlignment = rtfRight
        Case "Drawing"
            'ToDo: Add 'Drawing' button code.
    End Select
End Sub

Private Sub mnuHelpAbout_Click()
    frmAbout.Show vbModal, Me
End Sub

Private Sub mnuHelpSearchForHelpOn_Click()
    Dim nRet As Integer


    'if there is no helpfile for this project display a message to the user
```

```
    'you can set the HelpFile for your application in the
    'Project Properties dialog
    If Len(App.HelpFile) = 0 Then
        MsgBox "Unable to display Help Contents. There is no Help associated
with this project.", vbInformation, Me.Caption
    Else
        On Error Resume Next
        nRet = OSWinHelp(Me.hwnd, App.HelpFile, 261, 0)
        If Err Then
            MsgBox Err.Description
        End If
    End If

End Sub

Private Sub mnuHelpContents_Click()
    SetupFigure.Visible = True
    Exit Sub
    Dim nRet As Integer

    'if there is no helpfile for this project display a message to the user
    'you can set the HelpFile for your application in the
    'Project Properties dialog
    If Len(App.HelpFile) = 0 Then
        MsgBox "Unable to display Help Contents. There is no Help associated
with this project.", vbInformation, Me.Caption
    Else
        On Error Resume Next
        nRet = OSWinHelp(Me.hwnd, App.HelpFile, 3, 0)
        If Err Then
            MsgBox Err.Description
        End If
    End If

End Sub


Private Sub mnuWindowArrangeIcons_Click()
    Me.Arrange vbArrangeIcons
End Sub

Private Sub mnuWindowTileVertical_Click()
    Me.Arrange vbTileVertical
End Sub

Private Sub mnuWindowTileHorizontal_Click()
    Me.Arrange vbTileHorizontal
End Sub

Private Sub mnuWindowCascade_Click()
    Me.Arrange vbCascade
End Sub

Private Sub mnuWindowNewWindow_Click()
    LoadNewDoc
End Sub
```

```
Private Sub mnuToolsOptions_Click()
    frmOptions.Show vbModal, Me
End Sub

Private Sub mnuViewOptions_Click()
    frmOptions.Show vbModal, Me
End Sub

Private Sub mnuViewRefresh_Click()
    'ToDo: Add 'mnuViewRefresh_Click' code.
    MsgBox "Add 'mnuViewRefresh_Click' code."
End Sub

Private Sub mnuViewStatusBar_Click()
    mnuViewStatusBar.Checked = Not mnuViewStatusBar.Checked
    sbStatusBar.Visible = mnuViewStatusBar.Checked
End Sub

Private Sub mnuViewToolbar_Click()
    mnuViewToolbar.Checked = Not mnuViewToolbar.Checked
    tbToolBar.Visible = mnuViewToolbar.Checked
End Sub

Private Sub mnuEditPasteSpecial_Click()
    'ToDo: Add 'mnuEditPasteSpecial_Click' code.
    MsgBox "Add 'mnuEditPasteSpecial_Click' code."
End Sub

Private Sub mnuEditPaste_Click()
    On Error Resume Next
  '  ActiveForm.rtfText.SelRTF = Clipboard.GetText

End Sub

Private Sub mnuEditCopy_Click()
    On Error Resume Next
   ' Clipboard.SetText ActiveForm.rtfText.SelRTF

End Sub

Private Sub mnuEditCut_Click()
    On Error Resume Next
   ' Clipboard.SetText ActiveForm.rtfText.SelRTF
  '  ActiveForm.rtfText.SelText = vbNullString

End Sub

Private Sub mnuEditUndo_Click()
    'ToDo: Add 'mnuEditUndo_Click' code.
    MsgBox "Add 'mnuEditUndo_Click' code."
End Sub


Private Sub mnuFileExit_Click()
    ilonl Dev%, 0
    Close All
    End
```

```
End Sub

Private Sub mnuFileSend_Click()
    'ToDo: Add 'mnuFileSend_Click' code.
    MsgBox "Add 'mnuFileSend_Click' code."
End Sub

Private Sub mnuFilePrint_Click()
    On Error Resume Next
    If ActiveForm Is Nothing Then Exit Sub

    With dlgCommonDialog
        .DialogTitle = "Print"
        .CancelError = True
        .Flags = cdlPDReturnDC + cdlPDNoPageNums
     '   If ActiveForm.rtfText.SelLength = 0 Then
     '       .Flags = .Flags + cdlPDAllPages
     '   Else
     '       .Flags = .Flags + cdlPDSelection
     '   End If
        .ShowPrinter
'        If err <> MSComDlg.cdlCancel Then
'            ActiveForm.rtfText.SelPrint .hDC
'        End If
    End With

End Sub

Private Sub mnuFilePrintPreview_Click()
    'ToDo: Add 'mnuFilePrintPreview_Click' code.
    MsgBox "Add 'mnuFilePrintPreview_Click' code."
End Sub

Private Sub mnuFilePageSetup_Click()
    On Error Resume Next
    With dlgCommonDialog
        .DialogTitle = "Page Setup"
        .CancelError = True
        .ShowPrinter
    End With

End Sub

Private Sub mnuFileProperties_Click()
    'ToDo: Add 'mnuFileProperties_Click' code.
    MsgBox "Add 'mnuFileProperties_Click' code."
End Sub

Private Sub mnuFileSaveAll_Click()
    'ToDo: Add 'mnuFileSaveAll_Click' code.
    MsgBox "Add 'mnuFileSaveAll_Click' code."
End Sub

Private Sub mnuFileSaveAs_Click()
    If ActiveForm.Caption <> "Viraj2007 Multimeter" Then
        ActiveForm.SaveAs
    Else
```

```
        frmSR850.WindowState = 1
        ActiveForm.SaveAs
        frmSR850.WindowState = 0
    End If
End Sub

Private Sub mnuFileSave_Click()
    If ActiveForm.Caption <> "Viraj2007 Multimeter" Then
        ActiveForm.Save
    Else
        frmSR850.WindowState = 1
        ActiveForm.Save
        frmSR850.WindowState = 0
    End If
    End Sub

Private Sub mnuFileClose_Click()
    'ToDo: Add 'mnuFileClose_Click' code.
    MsgBox "Add 'mnuFileClose_Click' code."
End Sub

Private Sub mnuFileOpen_Click()
    LoadNewDoc
    ActiveForm.Fopen
End Sub

Private Sub mnuFileNew_Click()
    LoadNewDoc
End Sub

Public Sub multioff()
tbToolBar.Buttons.Item(5).Value = tbrUnpressed
End Sub

Private Sub txtStart_Change()
    On Error Resume Next
    ActiveForm.SetScale Val(txtStart.Text), Val(txtStop.Text)
End Sub

Private Sub txtStop_Change()
    On Error GoTo ee

    ActiveForm.SetScale txtStart, txtStop
ee:
End Sub
Sub UpdateDisp()
    Me.txtCurrPos = Format(gblCurWL&, "####.##")
End Sub

Sub UpdateDispData(data As Single)
    Me.txtData = ChangeUnit(data)
    Me.txtCurrPos = gblCurWL& / 100
    DoEvents
End Sub

Sub UpdateMainScreen()
```

```
    sbStatusBar.Panels(1).Text = "Current Grating :" & gblNGrtSelted& & "   " &
"Groove : " & gblCurGr& & " g/mm  " & "Blaze : " & gblCurBl& & " nm"
    txtCurrPos.Text = gblCurWL& / 100
    txtGrating.Text = gblNGrtSelted&
    txtFilter.Text = gblFilter%
    txtInputSlit.Text = gblS1%
    txtOutputSlit.Text = gblS2%
    If ShutOpen Then
        shpShutter.FillColor = &H8000000F
        lblShutter.Caption = "Open"
    Else
       shpShutter.FillColor = &H0
       lblShutter.Caption = "Closed"
    End If


End Sub
```

****

frmDocument.frm

****

```
'Copyrights - Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State
University

    Dim oBook As Excel.Workbook
    Dim oSheet As Excel.Worksheet
    Dim oChart As Excel.Chart
    Dim NoG As Long
    Dim DataBakGnd() As Variant
    Private FStep As Integer
    Private SStart As Single
    Private SStep As Single
    Private SStop As Single
    Private SLamda As Long
    Private LInten As Double 'Light Intensity W/cm²
    Dim n As Integer 'No of data points


Private Sub cmbDescription_Click()
    txtCellNo = cmbDescription.ListIndex + 1
    DescripChange
End Sub

Private Sub cmbDescription_DropDown()
    DescripChange
End Sub

Private Sub cmbDescription_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        DescripChange
    End If
End Sub

Private Sub cmbDescription_LostFocus()
    DescripChange
```

```
End Sub
Sub DescripChange()
    cmbDescription.List(txtCellNo - 1) = cmbDescription.Text
    cmbDescription.ListIndex = txtCellNo - 1
    With oSheet
    If NoG > 0 Then
        .Cells(3, (4 * NoG) - 2).Value = cmbDescription.List(NoG - 1)
    End If
    End With
End Sub

Private Sub Form_Load()
    cmbDescription.List(0) = "Vbias=        Rbias=        Temp="
    cmbDescription.ListIndex = 0
    txtCellNo = cmbDescription.ListIndex + 1

    OLE1.CreateEmbed "", "excel.chart"
    Set oBook = OLE1.object
    Set oChart = oBook.Charts(1)
    Set oSheet = oBook.Worksheets(1)

    oSheet.Cells.Clear
    oChart.ChartType = xlXYScatterSmoothNoMarkers
    oChart.SetSourceData oSheet.Range("A13:B114"), xlColumns
    With oChart

        .HasTitle = True
        .ChartTitle.AutoScaleFont = False
        .ChartTitle.Font.Size = 12
        .ChartTitle.Text = "Spectral Response of the Sample"
        .HasLegend = True
        .Legend.Interior.ColorIndex = xlNone
        .Legend.Border.LineStyle = xlNone
        .Legend.AutoScaleFont = False
        .Legend.Font.Size = 8
        .Legend.Left = 200
        .Legend.Top = 50
        .Axes(xlCategory, xlPrimary).HasTitle = True
        .Axes(xlCategory, xlPrimary).AxisTitle.AutoScaleFont = False
        .Axes(xlCategory, xlPrimary).AxisTitle.Font.Size = 12
        .Axes(xlCategory,        xlPrimary).AxisTitle.Characters.Text        =
"Wavelength(nm)"
        .Axes(xlValue, xlPrimary).HasTitle = True
        .Axes(xlValue,   xlPrimary).AxisTitle.Characters.Text   =   "Sample   Raw
Response(V)"
        .Axes(xlValue, xlPrimary).AxisTitle.AutoScaleFont = False
        .Axes(xlValue, xlPrimary).AxisTitle.Font.Size = 12
    End With
    With oChart.Axes(xlValue)
        .HasMajorGridlines = False
        .HasMinorGridlines = False
        .MajorTickMark = xlInside
        .MinorTickMark = xlNone
        .TickLabelPosition = xlNextToAxis
        .TickLabels.AutoScaleFont = False
'        .TickLabels.NumberFormat = "0.00"
        .TickLabels.Font.Size = 12
```

```
        .MinimumScale = 0
        '.MaximumScale = 1

    End With
    With oChart.Axes(xlCategory)
        .HasMajorGridlines = False
        .HasMinorGridlines = False
        .MajorTickMark = xlInside
        .MinorTickMark = xlNone
        .TickLabelPosition = xlNextToAxis
        .TickLabels.AutoScaleFont = False
 '      .TickLabels.NumberFormat = "0.00"
        .TickLabels.Font.Size = 12
        .MinimumScale = fMainForm.txtStart
        .MaximumScale = fMainForm.txtStop

    End With

    oChart.PlotArea.Interior.ColorIndex = xlNone

    OLE1.Visible = True
 '   OLE1.Close  'Deactivate the OLE container
    Application.Assistant.Visible = True
    If optAutoTrack(0).Value = True Then
        AutoTrack = True
    Else
        AutoTrack = False
    End If

End Sub

Private Sub Form_Resize()
    On Error Resume Next
    OLE1.Move 0, 1800, Me.ScaleWidth, Me.ScaleHeight - 2100
    tbsGtoS.Top = 1440
    tbsGtoS.Width = Me.ScaleWidth
    Frame2.Top = Me.ScaleHeight - 835
    Frame2.Width = Me.ScaleWidth
    sldpos.Width = Frame2.Width - 300

End Sub

Sub Fopen()
    Dim sFile As String
    With CommonDialog
        .DialogTitle = "Open"
        .CancelError = False
        .Filter = "Excel File (*.xls)|*.xls"
        .DialogTitle = "Open IR Action Spectrum File"
        .ShowOpen
        If Len(.FileName) = 0 Then
            Exit Sub
        End If
        sFile = .FileName
    End With
    Caption = sFile
```

```
    OLE1.CreateEmbed sFile, "excel.chart"
    Set oBook = OLE1.object
    Set oSheet = oBook.Worksheets(1)

    With oSheet
        If Left(.Cells(1, 1).Value, 39) <> "Viraj2008 UV-VI-IR monochromator
Data File" Then
            MsgBox "This is not a Viraj2008 UV-VI-IR monochromator Data file",
vbOKOnly + vbInformation, "Invalied file format"
            Exit Sub
        End If
        Set oChart = oBook.Charts(1)
        c = 1
        xx = .Cells(1, c).Value
        While xx <> ""
            c = c + 2
            xx = .Cells(1, c).Value
        Wend
    End With
    NoG = (c - 1) / 2
    For Z% = 1 To NoG
        cmbDescription.List(Z% - 1) = oSheet.Cells(3, 2 * Z%).Value
    Next Z%
    cmbDescription.AddItem "Sample No " & LTrim(Str(NoG + 1))
End Sub

Sub Save()
    Dim sFile As String
    If Left$(Caption, 13) = "Spectrum Book" Then
        On Error GoTo ErrHandler
        With CommonDialog
            .CancelError = True
            .Filter = "Excel File (*.xls)|*.xls"
            .DialogTitle = "Save IV Curve File"
            .Flags = cdlOFNOverwritePrompt
            .ShowSave
            If Len(.FileName) = 0 Then
                Exit Sub
            End If
            sFile = .FileName
        End With
        oBook.SaveCopyAs sFile
        Caption = sFile
    Else
        sFile = Caption
        oBook.SaveCopyAs sFile
    End If
ErrHandler:
    Exit Sub
End Sub

Sub SaveAs()
    Dim sFile As String
    On Error GoTo ErrHandler
    With CommonDialog
        .CancelError = True
        .Filter = "Excel File (*.xls)|*.xls"
```

```
        .DialogTitle = "Save IV Curve File As"
        .Flags = cdlOFNOverwritePrompt
        .ShowSave
        If Len(.FileName) = 0 Then
            Exit Sub
        End If
        sFile = .FileName
    End With
    oBook.SaveCopyAs sFile
    Caption = sFile

ErrHandler:
    Exit Sub
End Sub

Sub ScanSample(Min&, Step As Single, Max&, Delay As Integer, Avg As Integer)
    Dim TCDelay As Integer
    n = (Max& - Min&) / Step + 1 'total no of data

    TCDelay = CInt(TimeConsSec * 1000)
    If PauseStop = 2 Then
        PauseStop = 0
        Exit Sub
    End If
    PInc = 100 * Step / (Max - Min)
    sldpos.Value = 0
    fMainForm.tbToolBar.Enabled = False
    MousePointer = 11

    NoG = NoG + 1
    Cn% = (NoG * 7) - 6
    Rn% = 14

    MsgWait "Please Wait... Moving to Starting Point"
    DoEvents
    MsgWait ""

    GoToLamdaBG Min
    MsgWait ""
    With oSheet
        .Cells(1, Cn%).Value = "Viraj2008 UV-VI-IR monochromator Data File"
        .Cells(2, Cn%).Value = "Date/Time"
        .Cells(3, Cn%).Value = "Sample Description :"
        .Cells(3, Cn% + 1).Value = "Sample No " & NoG
        .Cells(4, Cn%).Value = "Chopping Frequency    ="
        .Cells(5, Cn%).Value = "Time Constant         ="
        .Cells(6, Cn%).Value = "Enterence Slit width (um)="
        .Cells(7, Cn%).Value = "Exit Slit width (um)="
        .Cells(8, Cn%).Value = "Delay before start averaging ="
        .Cells(9, Cn%).Value = "No of data to avarage ="
        .Cells(10, Cn%).Value = "Pre-amp gain          ="
        .Cells(11, Cn%).Value = "Sample Area          ="
        .Cells(12, Cn%).Value = "No of Data Points    ="

        .Cells(13, Cn%).Value = "Wavelength (nm)"
        .Cells(13, Cn% + 1).Value = "Sample Voltage Response(V)"
        .Cells(13, Cn% + 2).Value = "Gain Corrected Response(V)"
```

```
        .Cells(13, Cn% + 3).Value = "Current Response(A)"
        .Cells(13, Cn% + 4).Value = "Light Intensity (W/cm²)"
        .Cells(13, Cn% + 5).Value = "sample Responsivity (V/W)"
        .Cells(13, Cn% + 6).Value = "sample Responsivity (A/W)"
        .Columns.AutoFit

    End With

    Wait Delay
    SetGain 'if overload detected decrease the sensitivity
    With oSheet
        wl% = gblCurWL& / 100
        .Cells(Rn%, Cn%).Value = wl%
        .Cells(Rn%, Cn% + 1).Value = SR850AvgData(Avg, TCDelay)

        f = "=R[13]C[" & Trim(Str(Cn%)) & "]/" & Trim(txtPAGain.Text)
        .Cells(Rn%, Cn% + 2).Formula = f
        'Set Range0 = .Range("C14")
        Set Range1 = .Range(.Cells(Rn%, Cn% + 2), .Cells(Rn%, Cn% + 2))
        Set Range2 = .Range(.Cells(Rn%, Cn% + 2), .Cells(Rn% + n, Cn% + 2))
        'Range0.Copy Destination:=Range1
        Range1.AutoFill Destination:=Range2

       '.Cells(Rn%,   Cn%   +   2).Value   =   .Cells(Rn%,   Cn%   +   1).Value   *
Val(txtPAGain.Text)

       '.Cells(Rn%, Cn% + 3).Value = "?"
        .Cells(Rn%, Cn% + 4).Value = Intensity(wl%)

        f = "=R[13]C[" & Trim(Str(Cn% + 1)) & "]/(R[13]C[" & Trim(Str(Cn% + 3))
& "]*" & Trim(txtArea.Text) & ")"
        .Cells(Rn%, Cn% + 5).FormulaR1C1 = f
        Set Range1 = .Range(.Cells(Rn%, Cn% + 5), .Cells(Rn%, Cn% + 5))
        Set Range2 = .Range(.Cells(Rn%, Cn% + 5), .Cells(Rn% + n, Cn% + 5))
        Range1.AutoFill Destination:=Range2

        '.Cells(Rn%, Cn% + 5).Value = .Cells(Rn%, Cn% + 2).Value / (.Cells(Rn%,
Cn% + 4).Value * Val(txtArea.Text))


    End With
    With oChart
        .Axes(xlValue).MinimumScale = Auto
        If .Axes(xlCategory).MaximumScale < CInt(Max / 90) * 100 And Min < Max
Then
            .Axes(xlCategory).MaximumScale = CInt(Max / 90) * 100
            .Axes(xlCategory).MinimumScale = Min
        End If
    End With

    If NoG > 1 Then
        oChart.SeriesCollection.NewSeries
    End If

    With oSheet
        oChart.SeriesCollection(NoG).Name = oSheet.Range(oSheet.Cells(3, Cn% +
1), oSheet.Cells(3, Cn% + 1))
```

```
        oChart.SeriesCollection(NoG).XValues    =    .Range(.Cells(14,    Cn%),
.Cells(14 + n, Cn%))
        oChart.SeriesCollection(NoG).Values  =  .Range(.Cells(14,  Cn%  +  1),
.Cells(14 + n, Cn% + 1))
    End With

    DoEvents

    Rn% = Rn% + 1
    sldpos.Value = PInc
' *******************************************************
    For SLamda = Min + Step To Max Step Step
        If PauseStop = 2 Then
            Exit For
        End If
        GoToLamdaBG SLamda

        With oSheet
            wl% = gblCurWL& / 100
            .Cells(Rn%, Cn%).Value = wl%
            STime = Timer
            Wait Delay
            SetGain 'if overload detected decrease the sensitivity
            .Cells(Rn%, Cn% + 1).Value = SR850AvgData(Avg, TCDelay)
            lblRealDelay.Caption  =  Format((Timer  -  STime)  *  1000,  "#0000.00
ms")
            '.Cells(Rn%,  Cn%  +  2).Value  =  .Cells(Rn%,  Cn%  +  1).Value  *
Val(txtPAGain.Text)
           '.Cells(Rn%, Cn% + 3).Value = "?"
            .Cells(Rn%, Cn% + 4).Value = Intensity(wl%)
            '.Cells(Rn%,  Cn%  +  5).Value  =  .Cells(Rn%,  Cn%  +  2).Value  /
(.Cells(Rn%, Cn% + 4).Value * Val(txtArea.Text))

            'LInten = LightInten(gblCurWL&)
            'oSheet.Cells(Rn%, Cn% + 2).Value = LInten
            'to calculate responsivity (A/W)
            'oSheet.Cells(Rn%,  Cn%  +  3).Value  =  oSheet.Cells(Rn%,  Cn%  +
1).Value * 10000 / (LInten * txtArea.Text)
            'to calculate IPCE%
            'oSheet.Cells(Rn%,  Cn%  +  4).Value  =  oSheet.Cells(Rn%,  Cn%  +
3).Value * 125000 / gblCurWL&
            'oChart.SeriesCollection(2 * NoG).XValues = .Range(.Cells(14, Cn%),
.Cells(Rn%, Cn%))
            'oChart.SeriesCollection(2 * NoG).Values = .Range(.Cells(14, Cn% +
1), .Cells(Rn%, Cn% + 1))
            'oChart.SeriesCollection(NoG).XValues  =  .Range(.Cells(14,  Cn%),
.Cells(Rn%, Cn%))
            'oChart.SeriesCollection(NoG).Values = .Range(.Cells(14, Cn% + 1),
.Cells(Rn%, Cn% + 1))

            .Cells(12, Cn% + 1).Value = Rn% - 13
        End With

        Rn% = Rn% + 1
        DoEvents
        If sldpos.Value + PInc < 101 Then
            sldpos.Value = sldpos.Value + PInc
```

```
        End If
        WriteToSR850 "OUTP?3"
        fMainForm.UpdateDispData ReadSR850
        While PauseStop = 1
           DoEvents
        Wend
    Next SLamda
'    Wend
'******************************************************

'    oSheet.Cells(Rn%, Cn%).Value = 0
'    oSheet.Cells(Rn%, Cn% + 1).Value = 0
'    oChart.SeriesCollection(NoG).XValues = oSheet.Range(oSheet.Cells(14, Cn%), 
oSheet.Cells(Rn%, Cn%))
'     oChart.SeriesCollection(NoG).Values = oSheet.Range(oSheet.Cells(14, Cn% + 
1), oSheet.Cells(Rn%, Cn% + 1))

     With oSheet

        .Cells(2, Cn% + 1).Value = Date & " at " & Time
        .Cells(3, Cn% + 1).Value = cmbDescription.List(NoG - 1)

        WriteToSR850 "FREQ?"
        .Cells(4, Cn% + 1).Value = Format(ReadSR850, "#000.0")

        .Cells(5, Cn% + 1).Value = TimeConstant
        .Cells(6, Cn% + 1).Value = fMainForm.txtInputSlit.Text
        .Cells(7, Cn% + 1).Value = fMainForm.txtOutputSlit.Text
        .Cells(8, Cn% + 1).Value = fMainForm.txtDelay.Text
        .Cells(9, Cn% + 1).Value = fMainForm.txtAvg.Text
        .Cells(10, Cn% + 1).Value = Me.txtPAGain.Text
        .Cells(11, Cn% + 1).Value = Me.txtArea.Text
        .Cells(12, Cn% + 1).Value = Rn% - 14
    End With

    'cmbDescription.List((2 * NoG) - 2) = "Sample No " & LTrim(Str(NoG))

    cmbDescription.AddItem LTrim(cmbDescription.List(NoG - 1))
    cmbDescription.ListIndex = NoG - 1

    fMainForm.cmdPause.Enabled = False
    fMainForm.cmdStop.Enabled = False
    fMainForm.cmdPlay.Enabled = True

    MousePointer = 0
    If PauseStop = 0 Then sldpos.Value = sldpos.Max
    fMainForm.tbToolBar.Enabled = True

End Sub

Sub ScanABSBakGnd(Min&, Step As Single, Max&, Delay As Integer, Avg As Integer)
    ReDim DataBakGnd(1 To ((Max - Min) / Step) + 1, 1 To 2)
    Me.Caption = Me.Caption & "Scanning Background Spectra"
    Dim TCDelay As Integer
    TCDelay = CInt(TimeConsSec * 1000)
    MsgBox TCDelay
    If PauseStop = 2 Then
```

```
        PauseStop = 0
        Exit Sub
    End If
    PInc = 100 * Step / (Max - Min)
    sldpos.Value = 0
    fMainForm.tbToolBar.Enabled = False
    MousePointer = 11

    NoG = 1
    Cn% = 1
    Rn% = 14

    MsgWait "Please Wait... Moving to Starting Point"
    DoEvents
    GoToLamdaBG Min
    MsgWait ""
    With oSheet
        .Cells(1, Cn%).Value = "Viraj2008 UV-VI-IR monochromator Data File"
        .Cells(2, Cn%).Value = "Date/Time"
        .Cells(3, Cn%).Value = "Absorption Background"
        .Cells(3, Cn% + 1).Value = "Sample No " & NoG
        .Cells(4, Cn%).Value = "Chopping Frequency    ="
        .Cells(5, Cn%).Value = "Time Constant         ="
        .Cells(6, Cn%).Value = "Enterence Slit width (um)="
        .Cells(7, Cn%).Value = "Exit Slit width (um)="
        .Cells(8, Cn%).Value = "Delay before start averaging ="
        .Cells(9, Cn%).Value = "No of data to avarage ="
        .Cells(10, Cn%).Value = "Pre-amp gain          ="
        .Cells(11, Cn%).Value = "Sample Area          ="
        .Cells(12, Cn%).Value = "No of Data Points    ="

        .Cells(13, Cn%).Value = "Wavelength (nm)"
        .Cells(13, Cn% + 1).Value = "Background spectra"
        .Columns.AutoFit

    End With

    Wait Delay
    SetGain 'if overload detected decrease the sensitivity

    DataBakGnd(1, 1) = gblCurWL& / 100
    DataBakGnd(1, 2) = SR850AvgData(Avg, TCDelay)

    With oChart
        oSheet.Cells(Rn%, Cn%).Value = DataBakGnd(1, 1)
        oSheet.Cells(Rn%, Cn% + 1).Value = DataBakGnd(1, 2)
        .Axes(xlValue).MinimumScale = Auto
        If .Axes(xlCategory).MaximumScale < CInt(Max / 90) * 100 And Min < Max
Then
            .Axes(xlCategory).MaximumScale = CInt(Max / 90) * 100
            .Axes(xlCategory).MinimumScale = Min
        End If
    End With
    oChart.SeriesCollection(NoG).Name = oSheet.Range(oSheet.Cells(3, Cn% + 1),
oSheet.Cells(3, Cn% + 1))

    DoEvents
```

```
    Rn% = Rn% + 1
    sldpos.Value = PInc
' **************************************************
    For SLamda = Min + Step To Max Step Step
        If PauseStop = 2 Then
            Exit For
        End If
        GoToLamdaBG SLamda

        With oSheet
            Wait Delay
            SetGain 'if overload detected decrease the sensitivity

            DataBakGnd(Rn% - 13, 1) = gblCurWL& / 100
            DataBakGnd(Rn% - 13, 2) = SR850AvgData(Avg, TCDelay)
            .Cells(Rn%, Cn%).Value = DataBakGnd(Rn% - 13, 1)
            .Cells(Rn%, Cn% + 1).Value = DataBakGnd(Rn% - 13, 2)

            oChart.SeriesCollection(NoG).XValues   =   .Range(.Cells(14,   Cn%),
.Cells(Rn%, Cn%))
            oChart.SeriesCollection(NoG).Values  =  .Range(.Cells(14,  Cn%  +  1),
.Cells(Rn%, Cn% + 1))
            .Cells(12, Cn% + 1).Value = Rn% - 13
        End With
        Rn% = Rn% + 1
        DoEvents
        If sldpos.Value + PInc < 101 Then
            sldpos.Value = sldpos.Value + PInc
        End If
        WriteToSR850 "OUTP?3"
        fMainForm.UpdateDispData ReadSR850
        While PauseStop = 1
           DoEvents
        Wend
    Next SLamda

    With oSheet

       .Cells(2, Cn% + 1).Value = Date & " at " & Time
       .Cells(3, Cn% + 1).Value = cmbDescription.List(NoG - 1)

       WriteToSR850 "FREQ?"
       .Cells(4, Cn% + 1).Value = Format(ReadSR850, "#000.0")

       .Cells(5, Cn% + 1).Value = TimeConstant
       .Cells(6, Cn% + 1).Value = fMainForm.txtInputSlit.Text
       .Cells(7, Cn% + 1).Value = fMainForm.txtOutputSlit.Text
       .Cells(8, Cn% + 1).Value = fMainForm.txtDelay.Text
       .Cells(9, Cn% + 1).Value = fMainForm.txtAvg.Text
       .Cells(10, Cn% + 1).Value = Me.txtPAGain.Text
       .Cells(11, Cn% + 1).Value = Me.txtArea.Text
       .Cells(12, Cn% + 1).Value = Rn% - 14
    End With

    'cmbDescription.List((2 * NoG) - 2) = "Sample No " & LTrim(Str(NoG))
```

```
    cmbDescription.AddItem LTrim(cmbDescription.List(NoG - 1))
    cmbDescription.ListIndex = NoG - 1

    fMainForm.cmdPause.Enabled = False
    fMainForm.cmdStop.Enabled = False
    fMainForm.cmdPlay.Enabled = True

    MousePointer = 0
    If PauseStop = 0 Then sldpos.Value = sldpos.Max
    fMainForm.tbToolBar.Enabled = True


End Sub

Sub Calibrate(Min&, Step As Single, Max&, Delay As Integer, Avg As Integer)
    Dim SiData(200 To 1100) As Double
    Dim InGaAsData(500 To 2961) As Double
    Dim TCDelay As Integer
    n = (Max& - Min&) / Step + 1 'total no of data

    txtPAGain.Text = "100 uA/V"
    MsgBox "Please set the current amp gain to 100 uA/V"

    Me.Caption = Me.Caption & "Measuring output light intensity"
    TCDelay = CInt(TimeConsSec * 1000)
    If PauseStop = 2 Then
        PauseStop = 0
        Exit Sub
    End If

    PInc = 100 * Step / (Max - Min)
    sldpos.Value = 0
    fMainForm.tbToolBar.Enabled = False
    MousePointer = 11

    fn% = FreeFile()
    Open App.Path & "\SiSpeRes.dat" For Input As #fn%
    Input #fn%, a, b
    k% = 200
    While Not EOF(fn%)
        Input #fn%, a, SiData(k%)
        k% = k% + 1
    Wend
    Close #fn%

    fn% = FreeFile()
    Open App.Path & "\InGaAsSpeRes.dat" For Input As #fn%
    Input #fn%, a, b
    k% = 500
    While Not EOF(fn%)
        Input #fn%, a, InGaAsData(k%)
        k% = k% + 1
    Wend
    Close #fn%


    NoG = 1
```

```
    Cn% = 1
    Rn% = 14

    MsgWait "Please Wait... Moving to Starting Point"
    DoEvents
    MsgWait ""
    GoToLamdaBG Min
    MsgWait ""
    With oSheet
        .Cells(1, Cn%).Value = "Viraj2008 UV-VI-IR Intensity calibration Data
File"
        .Cells(2, Cn%).Value = "Date/Time"
        .Cells(3, Cn%).Value = "Calibration"
        .Cells(3, Cn% + 1).Value = "Sample No " & NoG
        .Cells(4, Cn%).Value = "Chopping Frequency    ="
        .Cells(5, Cn%).Value = "Time Constant         ="
        .Cells(6, Cn%).Value = "Enterence Slit width (um)="
        .Cells(7, Cn%).Value = "Exit Slit width (um)="
        .Cells(8, Cn%).Value = "Delay before start averaging ="
        .Cells(9, Cn%).Value = "No of data to avarage ="
        .Cells(10, Cn%).Value = "Pre-amp gain          ="
        .Cells(11, Cn%).Value = "Sample Area          ="
        .Cells(12, Cn%).Value = "No of Data Points    ="

        .Cells(13, Cn%).Value = "Wavelength (nm)"
        .Cells(13, Cn% + 1).Value = "Detector Voltage Response(V)"
        .Cells(13, Cn% + 2).Value = "Detector Current Response(A)"
        .Cells(13, Cn% + 3).Value = "Spectral Response (A/W)"
        .Cells(13, Cn% + 4).Value = "Light Power (W)"
        .Cells(13, Cn% + 5).Value = "Light Intensity (W/cm²)"
        .Columns.AutoFit

    End With

    Wait Delay
    SetGain 'if overload detected decrease the sensitivity

    With oSheet
        wl% = gblCurWL& / 100
        .Cells(Rn%, Cn%).Value = wl%
        .Cells(Rn%, Cn% + 1).Value = SR850AvgData(Avg, TCDelay)
        .Cells(Rn%, Cn% + 2).Value = .Cells(Rn%, Cn% + 1).Value * 0.0001
        If gblDetectorSelted = 1 Then
           .Cells(Rn%, Cn% + 3).Value = SiData(wl%)
            DArea = 0.1296 'Si Detector area in cm²
        Else
            .Cells(Rn%, Cn% + 3).Value = InGaAsData(wl%)
            DArea = 0.007853981634 'InGaAs Detector area in cm²
        End If
        .Cells(Rn%, Cn% + 4).Value = .Cells(Rn%, Cn% + 2).Value / .Cells(Rn%,
Cn% + 3).Value
        Intensity(wl%) = .Cells(Rn%, Cn% + 4).Value / DArea
        .Cells(Rn%, Cn% + 5).Value = Intensity(wl%)
    End With
    With oChart
        .Axes(xlValue).MinimumScale = Auto
```

```
        If .Axes(xlCategory).MaximumScale < CInt(Max / 90) * 100 And Min < Max
Then
            .Axes(xlCategory).MaximumScale = CInt(Max / 90) * 100
            .Axes(xlCategory).MinimumScale = Min
        End If
    End With
    With oSheet
        oChart.SeriesCollection(NoG).Name = oSheet.Range(oSheet.Cells(3, Cn% +
1), oSheet.Cells(3, Cn% + 1))
        oChart.SeriesCollection(NoG).XValues    =    .Range(.Cells(14,    Cn%),
.Cells(14 + n, Cn%))
        oChart.SeriesCollection(NoG).Values  =  .Range(.Cells(14,  Cn%  +  1),
.Cells(14 + n, Cn% + 1))
    End With
    DoEvents

    Rn% = Rn% + 1
    sldpos.Value = PInc
' *******************************************************
    For SLamda = Min + Step To Max Step Step
        If PauseStop = 2 Then
            Exit For
        End If
        GoToLamdaBG SLamda

        With oSheet
            Wait Delay
            SetGain 'if overload detected decrease the sensitivity
            wl% = Int(gblCurWL& / 100)
            .Cells(Rn%, Cn%).Value = gblCurWL& / 100
            .Cells(Rn%, Cn% + 1).Value = SR850AvgData(Avg, TCDelay)
            .Cells(Rn%, Cn% + 2).Value = .Cells(Rn%, Cn% + 1).Value * 0.0001
            If gblDetectorSelted = 1 Then
               .Cells(Rn%, Cn% + 3).Value = SiData(wl%)
                DArea = 0.1296 'Si Detector area in cm²
            Else
                .Cells(Rn%, Cn% + 3).Value = InGaAsData(wl%)
                DArea = 0.007853981634 'InGaAs Detector area in cm²
            End If
            .Cells(Rn%,  Cn%  +  4).Value  =  .Cells(Rn%,  Cn%  +  2).Value  /
.Cells(Rn%, Cn% + 3).Value
            Intensity(wl%) = .Cells(Rn%, Cn% + 4).Value / DArea
            .Cells(Rn%, Cn% + 5).Value = Intensity(wl%)

            'oChart.SeriesCollection(NoG).XValues  =  .Range(.Cells(14,  Cn%),
.Cells(Rn%, Cn%))
            'oChart.SeriesCollection(NoG).Values = .Range(.Cells(14, Cn% + 1),
.Cells(Rn%, Cn% + 1))
            .Cells(12, Cn% + 1).Value = Rn% - 13

        End With

        Rn% = Rn% + 1
        DoEvents
        If sldpos.Value + PInc < 101 Then
            sldpos.Value = sldpos.Value + PInc
        End If
```

```
        WriteToSR850 "OUTP?3"
        fMainForm.UpdateDispData ReadSR850
        While PauseStop = 1
           DoEvents
        Wend
    Next SLamda

    With oSheet

       .Cells(2, Cn% + 1).Value = Date & " at " & Time
       .Cells(3, Cn% + 1).Value = cmbDescription.List(NoG - 1)

       WriteToSR850 "FREQ?"
       .Cells(4, Cn% + 1).Value = Format(ReadSR850, "#000.0")

       .Cells(5, Cn% + 1).Value = TimeConstant
       .Cells(6, Cn% + 1).Value = fMainForm.txtInputSlit.Text
       .Cells(7, Cn% + 1).Value = fMainForm.txtOutputSlit.Text
       .Cells(8, Cn% + 1).Value = fMainForm.txtDelay.Text
       .Cells(9, Cn% + 1).Value = fMainForm.txtAvg.Text
       .Cells(10, Cn% + 1).Value = Me.txtPAGain.Text
       .Cells(11, Cn% + 1).Value = Me.txtArea.Text
       .Cells(12, Cn% + 1).Value = Rn% - 14
    End With

    'cmbDescription.List((2 * NoG) - 2) = "Sample No " & LTrim(Str(NoG))

    cmbDescription.AddItem LTrim(cmbDescription.List(NoG - 1))
    cmbDescription.ListIndex = NoG - 1

    fMainForm.cmdPause.Enabled = False
    fMainForm.cmdStop.Enabled = False
    fMainForm.cmdPlay.Enabled = True

    MousePointer = 0
    If PauseStop = 0 Then sldpos.Value = sldpos.Max
    fMainForm.tbToolBar.Enabled = True


End Sub

Private Sub optAutoTrack_Click(Index As Integer)
    If Index = 0 Then
        AutoTrack = True
    Else
        AutoTrack = False
    End If
End Sub

Private Sub tbsGtoS_Click()

    Select Case tbsGtoS.SelectedItem.Index
        Case 1
            With oChart
                .ChartTitle.Text = "Raw Response of the Sample"
                .Axes(xlValue,  xlPrimary).AxisTitle.Characters.Text  =  "Sample
Raw Response(V)"
```

```
            End With

            With oSheet
                If .Cells(3, 1).Value = "Calibration" Then
                    oChart.SeriesCollection(1).Values  =  .Range(.Cells(14,  2),
.Cells(14 + n, 2))
                Else
                    For k = 1 To NoG
                        cc% = (k * 7) - 6
                        oChart.SeriesCollection(k).Values  =  .Range(.Cells(14,
cc% + 1), .Cells(14 + n, cc% + 1))
                    Next k
                End If
            End With
        Case 2
            With oChart
                .ChartTitle.Text = "Responsivity of the Sample"
                .Axes(xlValue,      xlPrimary).AxisTitle.Characters.Text      =
"Responsivity (V/W)"
            End With

            With oSheet
                If oSheet.Cells(3, 1).Value = "Calibration" Then
                    oChart.SeriesCollection(1).Values  =  .Range(.Cells(14,  4),
.Cells(14 + n, 4))
                Else
                    For k = 1 To NoG
                        cc% = (k * 7) - 6
                        oChart.SeriesCollection(k).Values  =  .Range(.Cells(14,
cc% + 5), .Cells(14 + n, cc% + 5))
                    Next k
                End If
            End With

        Case 3
            With oChart
                .ChartTitle.Text = "Monochoromator Light Intensity"
                .Axes(xlValue,  xlPrimary).AxisTitle.Characters.Text  =  "Light
Intensity (W/cm²)"
            End With

            With oSheet
                If oSheet.Cells(3, 1).Value = "Calibration" Then
                    oChart.SeriesCollection(1).Values  =  .Range(.Cells(14,  6),
.Cells(14 + n, 6))
                Else
                    For k = 1 To NoG
                        cc% = (k * 7) - 6
                        oChart.SeriesCollection(k).Values  =  .Range(.Cells(14,
cc% + 4), .Cells(14 + n, cc% + 4))
                    Next k
                End If
            End With

    End Select

End Sub
```

```
Private Sub txtCellNo_Click()
    CommonDialog.ShowColor
    If NoG > 0 Then
        With oChart.SeriesCollection(Val(txtCellNo))
            .Border.Color = CommonDialog.Color
            .MarkerBackgroundColor = CommonDialog.Color
            .MarkerForegroundColor = 0
            .MarkerSize = 4
        End With
    End If
End Sub

Sub SetScale(Min As Single, Max As Single)
    oChart.Axes(xlCategory).MinimumScale = Min
    oChart.Axes(xlCategory).MaximumScale = Max
End Sub
```

****

frmMonitor.frm

****

```
'Copyrights - Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State
University

Dim nn As Integer
Private Sub chkOnOff_Click()
    'On Error Resume Next
    If chkOnOff.Value Then

        WriteToSR850 "SENS?"
        CmbGain.ListIndex = Int(ReadSR850())

        WriteToSR850 "OFLT?"
        cmbTimeConst.ListIndex = Int(ReadSR850())

        cmbGrating.ListIndex = gblNGrtSelted& - 1
        txtWLength.Text = gblCurWL& / 100
        Option1(gblFilter% - 1).Value = True
        txtSlit1.Text = gblS1%
        txtSlit2.Text = gblS2%
        cmdAutoGain.Enabled = False
        cmdAutoReserve.Enabled = False
        DoEvents
        Timer1.Enabled = True

    Else
        Timer1.Enabled = False
        cmdAutoGain.Enabled = True
        cmdAutoReserve.Enabled = True
        fMainForm.optMode(1).Value = True
    End If
End Sub


Private Sub CmbGain_Click()
    WriteToSR850 "SENS" & Trim(Str(CmbGain.ListIndex))
```

```
End Sub

Private Sub cmbGrating_Click()
    txtMsg.Text = "Please Wait... Changing the Grating"
    txtMsg.Visible = True
    DoEvents
    Status% = DKGrtsel%(cmbGrating.ListIndex + 1, fMainForm.comComm1, 500)
    txtMsg.Visible = False

    If Status% < 128 Then
      gblNGrtSelted& = cmbGrating.ListIndex + 1
    Else
      MsgBox "An error has occurred !", MB_ICONSTOP
    End If
    fMainForm.UpdateMainScreen
End Sub

Private Sub cmbTimeConst_Click()
    WriteToSR850 "OFLT" & Trim(Str(cmbTimeConst.ListIndex))
End Sub


Private Sub cmdAutoGain_Click()
    WriteToSR850 "AGAN"
End Sub

Private Sub cmdAutoPhase_Click()
    WriteToSR850 "APHS"
End Sub

Private Sub cmdAutoReserve_Click()
    WriteToSR850 "ARSV"
End Sub

Private Sub cmdAutoScale_Click()
    WriteToSR850 "ASCL"
End Sub

Private Sub cmdClsScreen_Click()
    ClsScreen
End Sub

Private Sub cmdGo_Click()
    txtMsg.Text = "Please Wait... Changing the Wavelength"
    txtMsg.Visible = True
    DoEvents
    GoToLamdaCG txtWLength.Text
    txtMsg.Visible = False
End Sub

Private Sub cmdGotoScanMode_Click()
    Me.Visible = False
    fMainForm.optMode(1).Value = True
End Sub

Private Sub cmdGoZero_Click()
    txtMsg.Text = "Please Wait... Changing the Wavelength"
```

```
    txtMsg.Visible = True
    DoEvents
    GoToLamdaCG 0
    txtMsg.Visible = False
End Sub

Private Sub cmdMotPow_Click()
    If MotPowON Then
        MotorPower "OFF"
        shpMPwLED.FillColor = &H0
    Else
        MotorPower "ON"
        shpMPwLED.FillColor = &HFF&
        txtMsg.Text = "Please wait Initilizing Stepper Motors"
        txtMsg.Visible = True
        DoEvents
        InitializeMM
        InitializeDM
        txtMsg.Visible = False
    End If
End Sub

Private Sub cmdRefRatSet_Click()
Timer1.Interval = Val(txtRefrate.Text)
End Sub

Private Sub cmdSet_Click()
    SetSlt 2, Val(txtSlit2.Text)
    On Error GoTo cmdCanOKErr

    txtMsg.Text = "Please Wait... Changing the Slit Width"
    txtMsg.Visible = True
    DoEvents
    Status% = DKSltAdjCal%(31, Val(txtSlit1.Text), fMainForm.comComm1, 500)
    If Status% > 127 Then GoTo cmdCanOKErr
    gblS1% = Val(txtSlit1.Text)  ' S1 adjustment.
    gblOkCancel% = IDOK

    Status% = DKSltAdjCal%(32, Val(txtSlit2.Text), fMainForm.comComm1, 500)
    If Status% > 127 Then GoTo cmdCanOKErr
    gblS2% = Val(txtSlit2.Text)  ' S2 adjustment.
    gblOkCancel% = IDOK

    fMainForm.UpdateMainScreen
    txtMsg.Visible = False
    Exit Sub

cmdCanOKErr:
  MsgBox "Error code is : " & Err, MB_ICONEXCLAMATION

End Sub

Private Sub Form_Activate()
    ClsScreen
    optLamp(gblLightSelted - 1).Value = True
    optDetector(gblDetectorSelted - 1).Value = True
    WriteToSR850 "SENS?"
```

```
    CmbGain.ListIndex = Int(ReadSR850())

    WriteToSR850 "OFLT?"
    cmbTimeConst.ListIndex = Int(ReadSR850())

    cmbGrating.ListIndex = gblNGrtSelted& - 1
    txtWLength.Text = gblCurWL& / 100
    Option1(gblFilter% - 1).Value = True
    txtSlit1.Text = gblS1%
    txtSlit2.Text = gblS2%

End Sub

Private Sub Form_Resize()
    On Error Resume Next
    MSChart1.Move 1080, -140, Me.ScaleWidth - 900, Me.ScaleHeight - 1600
    Frame1.Top = Me.ScaleHeight - 2000
    Frame1.Left = 1350
    Frame5.Height = Me.ScaleHeight - 60
    txtMsg.Top = MSChart1.Height / 2 - 262
    txtMsg.Left = Me.ScaleWidth / 2 - 1767
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Timer1.Enabled = False
    fMainForm.optMode(1).Value = True
    'SaveFormData Me, App.Path & "\Settings_Monitor.dat"
End Sub

Private Sub frmShutter_Click()
    If shpShutter.FillColor = &H8000000F Then
        Shutter "close"
        lblShutter.Caption = "Closed"
        shpShutter.FillColor = &H0
    Else
        Shutter "open"
        lblShutter.Caption = "Open"
        shpShutter.FillColor = &H8000000F
    End If
End Sub

Private Sub optDetector_Click(Index As Integer)
    txtMsg.Text = "Please wait... Changing the Detector"
    txtMsg.Visible = True
    DoEvents
    RunDM frmStpMotSet.txtDetStp(Index)
    gblDetectorSelted = Index + 1
    txtMsg.Visible = False
    MsgBox "Please connect the Detector " & Trim(Str(Index + 1)) & " to the
pre-amp"
End Sub

Private Sub Option1_Click(Index As Integer)
    On Error GoTo cmdCanOKErr
    txtMsg.Text = "Please wait... Changing to Filter # " & Index + 1
    txtMsg.Visible = True
    DoEvents
```

```
    Status% = DKFilter%(Index + 1, fMainForm.comComm1, 5) '5: 5s Timeout%

    If Status% < 128 Then gblFilter% = Index + 1
    gblOkCancel% = IDOK
    fMainForm.UpdateMainScreen
    txtMsg.Visible = False
    Exit Sub

cmdCanOKErr:
  MsgBox "Error code is : " & Err, MB_ICONEXCLAMATION


End Sub

Private Sub optLamp_Click(Index As Integer)
    txtMsg.Text = "Please wait... Changing the light Source"
    txtMsg.Visible = True
    DoEvents
    RunMM frmStpMotSet.txtLmpStp(Index)
    gblLightSelted = Index + 1
    txtMsg.Visible = False
End Sub


Private Sub Timer1_Timer()
    Dim Dt As Single
    Dim SB As Byte
    'Static nn As Integer
    If nn >= 200 Or nn < 1 Then nn = 1
    Dt = SR850Data
    txtData.Text = ChangeUnit(Dt)
    With MSChart1
        .Row = nn
        .Column = 1
        .data = Dt
        nn = nn + 1
        For jj = nn To 200
            .Row = jj
            .Column = 1
            .data = Dt
        Next jj

    End With
    WriteToSR850 "LIAS?"
    SB = ReadSR850
    For k = 0 To 2
        If  (SB  And  2  ^  k)  =  2  ^  k  Then  lblLED(k).Visible  =  True  Else
lblLED(k).Visible = False
    Next k
End Sub

Private Sub ClsScreen()
    For k = 1 To 200
        With MSChart1
            .Row = k
            .Column = 1
```

```
            .data = " "

        End With
    Next k
nn = 1
End Sub

Sub SetSlt(SlitNo As Integer, SlitWidth As Integer)
End Sub
```

****

frmAdvSetup.frm

****

```
'Copyrights - Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State
University

Private Sub cmdClose_Click()
    Me.Visible = False
End Sub

Private Sub cmdRstDft_Click()
    ReadFormData Me, App.Path & "\Settings_Dflt_AdvSetup.dat"
    SaveFormData Me, App.Path & "\Settings_AdvSetup.dat"
End Sub

Private Sub cmdSave_Click()
    nl = Chr(10) & Chr(13)
    cho = MsgBox("You are attempting to change an important system setting." &
nl & "This may result incorect spectral measurment data." & nl & "Do you really
want save these new settings", vbExclamation + vbYesNo, "!!! WARNING !!!")
    If cho = 6 Then
       SaveFormData Me, App.Path & "\Settings_AdvSetup.dat"
    End If
End Sub

Private Sub Form_Load()
    ReadFormData Me, App.Path & "\Settings_AdvSetup.dat"
End Sub


Private Sub hscPic_Change()
    imgPic.Left = -hscPic.Value
End Sub

Private Sub smdStMoSett_Click()
    frmStpMotSet.Visible = True
End Sub
```

****

frmChopper.frm

****

```
'Copyrights - Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State
University
```

```
Private MidFre As Single

Private Sub cmdChFre_Click(Index As Integer)

End Sub

Private Sub cmdOk_Click()

    Unload Me
End Sub

Private Sub cmdSetFre_Click()
    Me.MousePointer = 11
    OutV = Val(txtChoFre.Text) / 33
    WriteToSR850 "AUXV1," & Trim(Str(OutV))

    While Abs(Val(txtChoFre.Text) - Val(lblDispChopperFre.Caption)) > 1
        If Val(txtChoFre.Text) > Val(lblDispChopperFre.Caption) Then
            OutV = OutV + 0.05
            WriteToSR850 "AUXV1," & Trim(Str(OutV))
            Wait 2000
            WriteToSR850 "FREQ?"
            lblDispChopperFre.Caption = Format(ReadSR850, "#000.00 Hz")
        Else
            OutV = OutV - 0.048
            WriteToSR850 "AUXV1," & Trim(Str(OutV))
            Wait 2000
            WriteToSR850 "FREQ?"
            lblDispChopperFre.Caption = Format(ReadSR850, "#000.00 Hz")
        End If
        DoEvents
    Wend

    While Abs(Val(txtChoFre.Text) - Val(lblDispChopperFre.Caption)) > 0.1
        If Val(txtChoFre.Text) > Val(lblDispChopperFre.Caption) Then
            OutV = OutV + 0.01
            WriteToSR850 "AUXV1," & Trim(Str(OutV))
            Wait 2000
            WriteToSR850 "FREQ?"
            lblDispChopperFre.Caption = Format(ReadSR850, "#000.00 Hz")
        Else
            OutV = OutV - 0.009
            WriteToSR850 "AUXV1," & Trim(Str(OutV))
            Wait 2000
            WriteToSR850 "FREQ?"
            lblDispChopperFre.Caption = Format(ReadSR850, "#000.00 Hz")
        End If
        DoEvents
    Wend
    While Abs(Val(txtChoFre.Text) - Val(lblDispChopperFre.Caption)) > 0.04
        If Val(txtChoFre.Text) > Val(lblDispChopperFre.Caption) Then
            OutV = OutV + 0.001
            WriteToSR850 "AUXV1," & Trim(Str(OutV))
            Wait 2000
            WriteToSR850 "FREQ?"
            lblDispChopperFre.Caption = Format(ReadSR850, "#000.00 Hz")
```

```
        Else
            OutV = OutV - 0.001
            WriteToSR850 "AUXV1," & Trim(Str(OutV))
            Wait 2000
            WriteToSR850 "FREQ?"
            lblDispChopperFre.Caption = Format(ReadSR850, "#000.00 Hz")
        End If
        DoEvents
    Wend
    Me.MousePointer = 0

End Sub

Private Sub Form_Load()
    WriteToSR850 "AUXM1,0"
    WriteToSR850 "FREQ?"
    lblDispChopperFre.Caption = Format(ReadSR850, "#000.00 Hz")
    txtChoFre.Text = lblDispChopperFre.Caption
End Sub

Private Sub optChoFre_Click(Index As Integer)
    Select Case Index

        Case 0
            WriteToSR850 "AUXV1,0.28"
            lblDispChopperFre.Caption = Format(7, "#000.00 Hz")
        Case 1
            WriteToSR850 "AUXV1,0.46"
            lblDispChopperFre.Caption = Format(13, "#000.00 Hz")
        Case 2
            WriteToSR850 "AUXV1,0.58"
            lblDispChopperFre.Caption = Format(17, "#000.00 Hz")
        Case 3
            WriteToSR850 "AUXV1,0.86"
            lblDispChopperFre.Caption = Format(27, "#000.00 Hz")
        Case 4
            WriteToSR850 "AUXV1,1.15"
            lblDispChopperFre.Caption = Format(37, "#000.00 Hz")
        Case 5
            WriteToSR850 "AUXV1,2.135"
            lblDispChopperFre.Caption = Format(71, "#000.00 Hz")

    End Select

End Sub
```

****

frmStpMotSet.frm

****

```
'Copyrights - Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State
University

Private Sub cmdDDec_Click(Index As Integer)
    txtDetStp(Index).Text = Val(txtDetStp(Index).Text) - 1
    RunDM Val(txtDetStp(Index).Text)
```

```
End Sub

Private Sub cmdDec_Click(Index As Integer)
    txtLmpStp(Index).Text = Val(txtLmpStp(Index).Text) - 1
    RunMM Val(txtLmpStp(Index).Text)
End Sub

Private Sub cmdDInc_Click(Index As Integer)
    txtDetStp(Index).Text = Val(txtDetStp(Index).Text) + 1
    RunDM Val(txtDetStp(Index).Text)
End Sub

Private Sub cmdDSet_Click(Index As Integer)
    RunDM Val(txtDetStp(Index).Text)
End Sub

Private Sub cmdExit_Click()
    Me.Visible = False
End Sub

Private Sub cmdInc_Click(Index As Integer)
    txtLmpStp(Index).Text = Val(txtLmpStp(Index).Text) + 1
    RunMM Val(txtLmpStp(Index).Text)
End Sub

Private Sub cmdRestDflt_Click()
    ReadFormData Me, App.Path & "\Settings_Dflt_StpMotSet.dat"
    SaveFormData Me, App.Path & "\Settings_StpMotSet.dat"
End Sub

Private Sub cmdSave_Click()
    nl = Chr(10) & Chr(13)
    cho = MsgBox("You are attempting to change an important system setting." &
nl & "This may result incorect Lamp and Detector positions" & nl & "Do you
really want save these new settings", vbExclamation + vbYesNo, "!!! WARNING
!!!")
    If cho = 6 Then
       SaveFormData Me, App.Path & "\Settings_StpMotSet.dat"
    End If

End Sub

Private Sub cmdSet_Click(Index As Integer)
    RunMM Val(txtLmpStp(Index).Text)
End Sub

Private Sub Form_Load()
    ReadFormData Me, App.Path & "\Settings_StpMotSet.dat"
End Sub

Private Sub Label11_Click()

End Sub
```

****

frmSR850.frm

```
**********************************************************************
'Copyrights - Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State
University

Private MP As String
Private Sub CmbGain_Click()
    WriteToSR850 "SENS" & Trim(Str(CmbGain.ListIndex))
End Sub

Private Sub cmbTimeConst_Click()
    WriteToSR850 "OFLT" & Trim(Str(cmbTimeConst.ListIndex))
End Sub

Private Sub cmdAutoGain_Click()
    WriteToSR850 "AGAN"
    'Wait 5000
    'WriteToSR850 "SENS ?"
    'CmbGain.ListIndex = ReadSR850
    'WriteToSR850 "OFLT ?"
    'cmbTimeConst.ListIndex = ReadSR850

End Sub

Private Sub cmdAutoPhase_Click()
    WriteToSR850 "APHS"
End Sub

Private Sub cmdAutoRese_Click()
    WriteToSR850 "ARSV"
End Sub

Private Sub cmdAutoScale_Click()
    WriteToSR850 "ASCL"
End Sub

Private Sub cmdAUX2_Click()
    If Me.lblAUX2.BackColor = &H0& Then
        Me.lblAUX2.BackColor = &HFF00&
        Me.txtAUX2.Enabled = True
        Me.sldAUX2.Enabled = True
        WriteToSR850 "AUXM2,0 "
        WriteToSR850 "AUXV2,0.000 "
    Else
        Me.lblAUX2.BackColor = &H0&
        WriteToSR850 "AUXV2,0.000 "
        Me.txtAUX2 = "0.000"
        Me.txtAUX2.Enabled = False
        Me.sldAUX2.Enabled = False
    End If

End Sub

Private Sub cmdAUX3_Click()
    If Me.lblAUX3.BackColor = &H0& Then
        Me.lblAUX3.BackColor = &HFF00&
        Me.txtAUX3.Enabled = True
```

```
        Me.sldAUX3.Enabled = True
        WriteToSR850 "AUXM3,0 "
        WriteToSR850 "AUXV3,0.000 "
    Else
        Me.lblAUX3.BackColor = &H0&
        WriteToSR850 "AUXV3,0.000 "
        Me.txtAUX3 = "0.000"
        Me.txtAUX3.Enabled = False
        Me.sldAUX3.Enabled = False
    End If
End Sub

Private Sub cmdAUX4_Click()
    If Me.lblAUX4.BackColor = &H0& Then
        Me.lblAUX4.BackColor = &HFF00&
        Me.txtAUX4.Enabled = True
        Me.sldAUX4.Enabled = True
        WriteToSR850 "AUXM4,0"
        WriteToSR850 "AUXV4,0.000"
    Else
        Me.lblAUX4.BackColor = &H0&
        WriteToSR850 "AUXV4,0.000"
        Me.txtAUX4 = "0.000"
        Me.txtAUX4.Enabled = False
        Me.sldAUX4.Enabled = False
    End If
End Sub

Private Sub cmdIMode_Click()
    WriteToSR850 "ISRC2"
    MP = "A"
End Sub

Private Sub cmdOk_Click()
    Me.Timer1.Enabled = False
    Unload Me
End Sub

Private Sub cmdRefresh_Click()
    If Me.cmdRefresh.Caption = "Refresh ON" Then
        Me.cmdRefresh.Caption = "Refresh OFF"
        Timer1.Enabled = True
    Else
        Me.cmdRefresh.Caption = "Refresh ON"
        Timer1.Enabled = False
    End If
End Sub

Private Sub cmdVMode_Click()
    WriteToSR850 "ISRC0"
    MP = "V"
End Sub

Private Sub Form_Load()
    WriteToSR850 "RMOD1"
    WriteToSR850 "SENS?"
    CmbGain.ListIndex = ReadSR850
```

```
    WriteToSR850 "OFLT?"
    cmbTimeConst.ListIndex = ReadSR850
    WriteToSR850 "AUXM2,0"

    WriteToSR850 "AUXV?2"
    Me.txtAUX2 = ReadSR850
    If Me.txtAUX2 <> 0 Then
        Me.txtAUX2.Enabled = True
        Me.sldAUX2.Enabled = True
        Me.lblAUX2.BackColor = &HFF00&
    End If

    WriteToSR850 "AUXV?3"
    Me.txtAUX3 = ReadSR850
    If Me.txtAUX3 <> 0 Then
        Me.txtAUX3.Enabled = True
        Me.sldAUX3.Enabled = True
        Me.lblAUX3.BackColor = &HFF00&
    End If

    WriteToSR850 "AUXV?4"
    Me.txtAUX4 = ReadSR850
    If Me.txtAUX4 <> 0 Then
        Me.txtAUX4.Enabled = True
        Me.sldAUX4.Enabled = True
        Me.lblAUX4.BackColor = &HFF00&
    End If
    WriteToSR850 "ISRC?"
    If ReadSR850 = 2 Then MP = "A" Else MP = "V"

    Timer1.Enabled = True
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Timer1.Enabled = False
End Sub

Private Sub sldAUX2_Click()
    Me.txtAUX2 = Me.sldAUX2.Value / 1000
End Sub

Private Sub sldAUX3_Click()
    Me.txtAUX3 = Me.sldAUX3.Value / 1000
End Sub

Private Sub sldAUX4_Click()
    Me.txtAUX4 = Me.sldAUX4.Value / 1000
End Sub

Private Sub Timer1_Timer()
    WriteToSR850 "OUTP?3"
    Me.lblR = ChangeUnit(ReadSR850) & MP
    WriteToSR850 "OUTP?1"
    Me.lblX = ChangeUnit(ReadSR850) & MP
    WriteToSR850 "OUTP?2"
    Me.lblY = ChangeUnit(ReadSR850) & MP
    WriteToSR850 "OUTP?4"
```

```
    Me.lblTheeta = Format(ReadSR850, "000.00")
End Sub

Private Sub txtAUX2_Change()
    If Val(txtAUX2.Text) > 10.5 Then txtAUX2.Text = 10.5
    If Val(txtAUX2.Text) < -10.5 Then txtAUX2.Text = -10.5
    Me.sldAUX2.Value = Val(Me.txtAUX2) * 1000
    WriteToSR850 "AUXV2," & Me.txtAUX2
End Sub

Private Sub txtAUX3_Change()
    If Val(txtAUX3.Text) > 10.5 Then txtAUX3.Text = 10.5
    If Val(txtAUX3.Text) < -10.5 Then txtAUX3.Text = -10.5
    Me.sldAUX3.Value = Val(Me.txtAUX3) * 1000
    WriteToSR850 "AUXV3," & Me.txtAUX3
End Sub

Private Sub txtAUX4_Change()
    If Val(txtAUX4.Text) > 10.5 Then txtAUX4.Text = 10.5
    If Val(txtAUX4.Text) < -10.5 Then txtAUX4.Text = -10.5
    Me.sldAUX4.Value = Val(Me.txtAUX4) * 1000
    WriteToSR850 "AUXV4," & Trim(Me.txtAUX4)
End Sub
```

B.3 Noise spectrum analyzer software

The detector noise characterization system is controlled using an in-house software package developed with Microsoft Visual Basic. The system is based on an SR785, dual channel, Fourier transform, dynamic signal analyzer and SR550, low noise preamplifier. The software uses the GPIB interface to communicate with the instrument. The software capabilities include: control all the front panel functions (except power on) through the software, read data from the SR785, record data in an embedded Microsoft Excel workbook, with single channel or dual channel modes.

B.3.1 Instrument configuration of the noise measurement setup

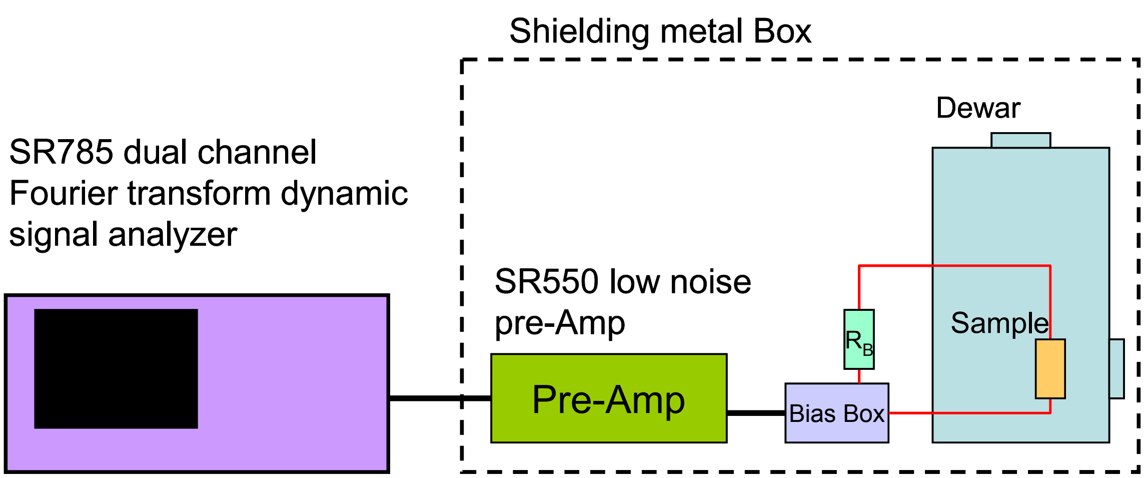


Figure B.21 Block diagram of the noise measurement setup. The sample, a power supply (battery), a bias resistor, and a low noise preamplifier are placed inside the noise-shielding cage. The sample is mounted in the same dewar system which is used for spectral measurements. A thick copper plate at the sample temperature is used as the radiation block to provide dark conditions for the detector.

B.3.2 User interfaces

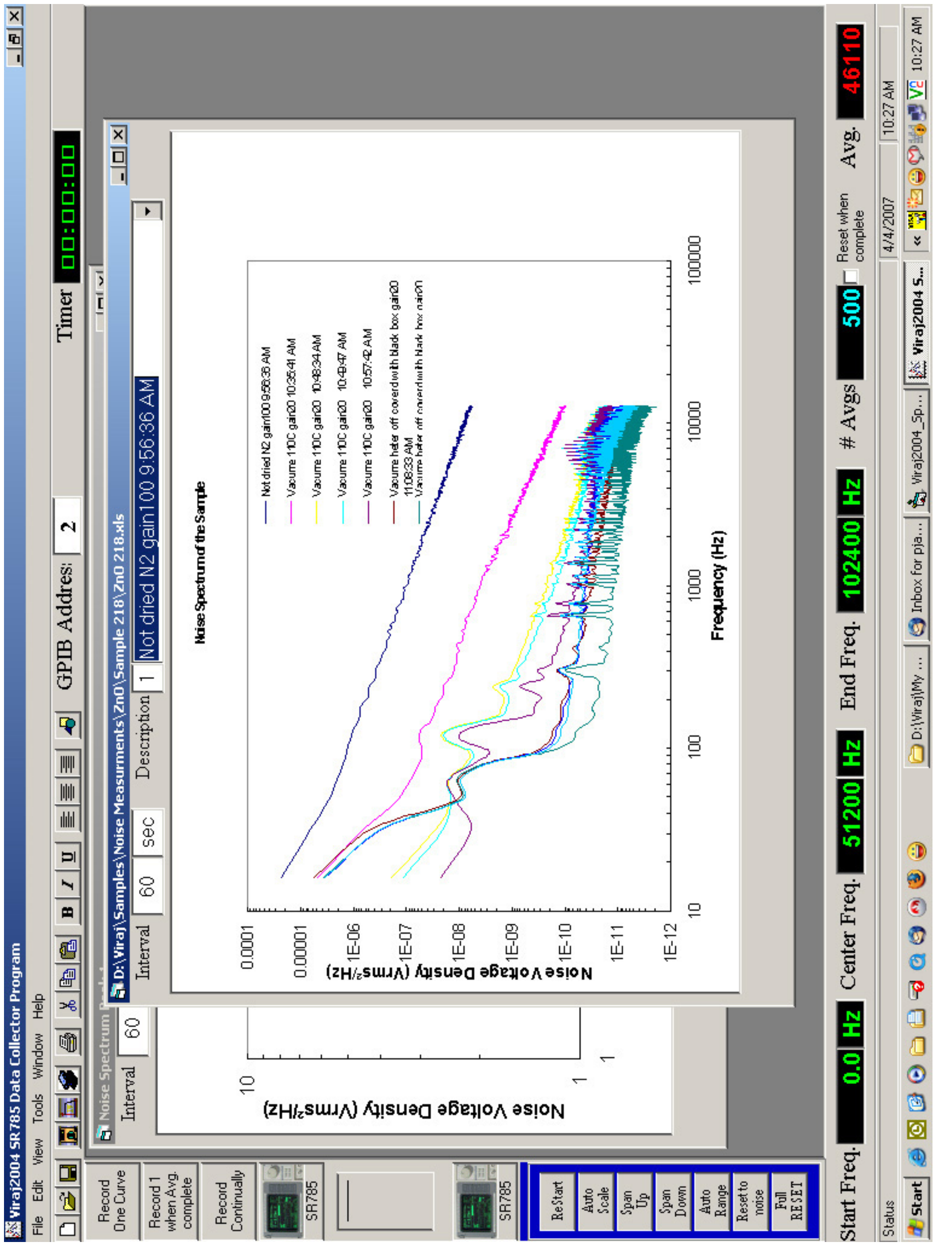


Figure B.22 The main user interface for the noise spectrum analyzer software showing real experimental data taken from the samples.

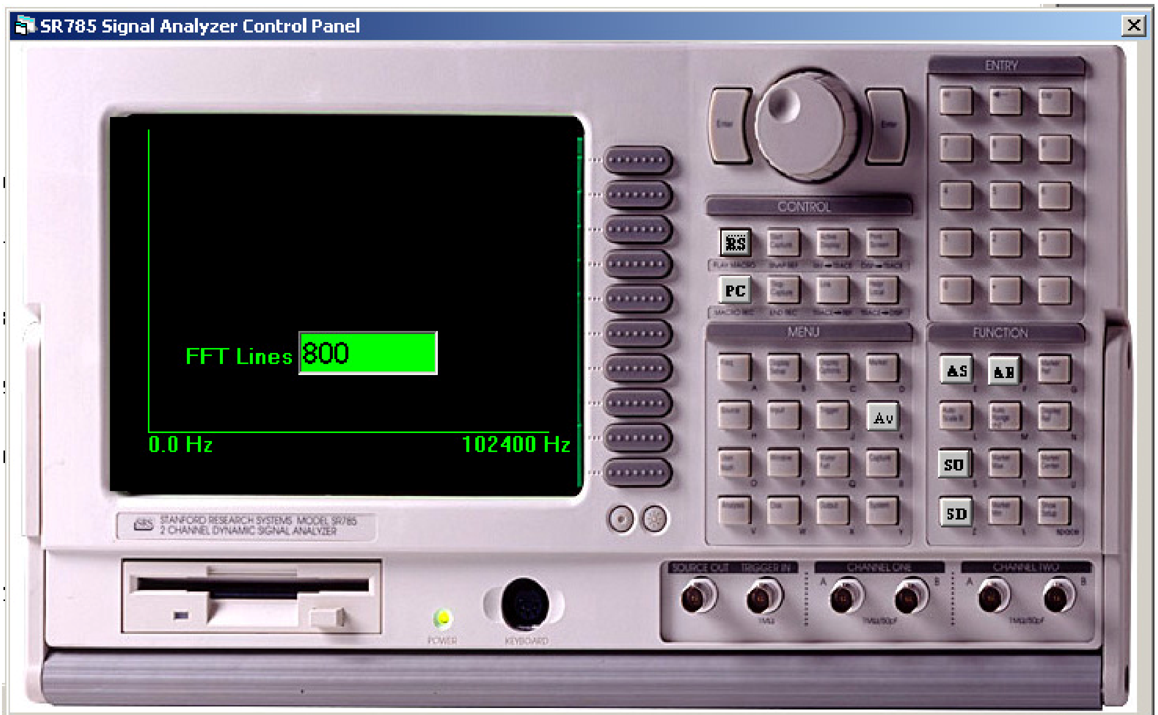


Figure B.23 Spectrum analyzer control panel. On screen buttons allow the user to change the parameters remotely without using the instrument front panel buttons.

B.3.3 Source code

```
*********************************************************************
SR785_GPIB.bas
*********************************************************************
'Copyrights - Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State
University

Attribute VB_Name = "SR785_GPIB"
Const BDINDEX = 0                    ' Board Index
Const PRIMARY_ADDR_OF_DMM = 10       ' Primary address of device
Const NO_SECONDARY_ADDR = 0          ' Secondary address of device
Const TIMEOUT = T10s                 ' Timeout value = 10 seconds
Const EOTMODE = 1                    ' Enable the END message
Const EOSMODE = 0                    ' Disable the EOS mode

Const ARRAYSIZE = 1024               ' Size of read buffer

Dim Msg As String * 100
Dim Dev As Integer
Dim ErrorMnemonic
Dim ValueStr As String * ARRAYSIZE
Dim data As String * 13700
Dim Response As Integer

'To Queries SR785 settings
Function QueriesSR785(CmdString As String)
    ilwrt Dev%, CmdString, Len(CmdString)
    If (ibsta And EERR) Then
        Call GPIBCleanup("Unable to write to device")
    End If

    ilrd Dev%, ValueStr, Len(ValueStr)
    If (ibsta And EERR) Then
        Call GPIBCleanup("Unable to read from device")
    End If
    QueriesSR785 = Val(Left$(ValueStr, ibcntl - 1))

End Function

'To write command to SR785
Sub WriteToSR785(CommandString As String)
    ilwrt Dev%, CommandString, Len(CommandString)
    If (ibsta And EERR) Then
        Call GPIBCleanup("Unable to write to device")
    End If
End Sub

'To read Data from SR785
Function ReadSR785() As String
    ilrd Dev%, data, Len(data) 'len(ValueStr)
    If (ibsta And EERR) Then
        Call GPIBCleanup("Unable to read from device")
    End If
    ReadSR785 = Trim(data)
```

```
End Function

Sub InitializeGPIB()
    Dev% = ildev(BDINDEX, PRIMARY_ADDR_OF_DMM, NO_SECONDARY_ADDR, _
                 TIMEOUT, EOTMODE, EOSMODE)
    If (ibsta And EERR) Then
        ErrMsg = "Unable to open device" & Chr(13) & "ibsta = &H" & _
        Chr(13) & Hex(ibsta) & "iberr = " & iberr
        MsgBox ErrMsg, vbCritical, "Error"
        End
    End If
    ' The application resets the GPIB portion of the device by calling
    ' ilclr.
    ilclr Dev%
    If (ibsta And EERR) Then
        Call GPIBCleanup("Unable to clear device")
    End If

End Sub

Sub GPIBCleanup(Msg$)

    ' After each GPIB call, the application checks whether the call
    ' succeeded. If an NI-488.2 call fails, the GPIB driver sets the
    ' corresponding bit in the global status variable. If the call
    ' failed, this procedure prints an error message, takes the device
    ' offline and exits.

    ErrorMnemonic = Array("EDVR", "ECIC", "ENOL", "EADR", "EARG", _
                          "ESAC", "EABO", "ENEB", "EDMA", "", _
                          "EOIP", "ECAP", "EFSO", "", "EBUS", _
                          "ESTB", "ESRQ", "", "", "", "ETAB")

    ErrMsg$ = Msg$ & Chr(13) & "ibsta = &H" & Hex(ibsta) & Chr(13) _
              & "iberr = " & iberr & " <" & ErrorMnemonic(iberr) & ">"
    MsgBox ErrMsg$, vbCritical, "Error"
    ilonl Dev%, 0

End Sub

Sub ResetSR785ToNoise()

    If Channel = 0 Then
        WriteToSR785 "DFMT 0" 'Set display format to single
        WriteToSR785 "MEAS 0,2" 'Set measurment Power Spectrum 1
    Else
        WriteToSR785 "DFMT 1" 'Set display format to Dual
        WriteToSR785 "MEAS 0,2" 'Set measurment Power Spectrum 1
        WriteToSR785 "MEAS 1,3" 'Set measurment Power Spectrum 2
    End If

    WriteToSR785 "LINK 1" 'Set Analyzer configeration to Dual channel
    WriteToSR785 "ACDT 0" 'Set Active display to A
    WriteToSR785 "MGRP " & Channel & " ,0" 'Set measurment group to FFT
    WriteToSR785 "VIEW " & Channel & " ,0" 'Set View of display A to Log
magnitude
    WriteToSR785 "UNDB " & Channel & " ,0" 'Set dB units Off
```

```
    WriteToSR785 "UNPK " & Channel & " ,2" 'Set pk units to rms
    WriteToSR785 "PSDU " & Channel & " ,1" 'Set psd units ON
    WriteToSR785 "XAXS " & Channel & " ,1" 'Set X axis scale type to Logarthmic
    WriteToSR785 "ISRC 0" 'Set Input source Analog
    WriteToSR785 "I1MD 0" 'Set Channel 1 input mode single-ended
    WriteToSR785 "I1GD 0" 'Set Channel 1 Input Grounding to Float
    WriteToSR785 "I1CP 1" 'Set Channel 1 Input Coupling to AC
    WriteToSR785 "I1AR 1" 'Set Channel 1 Auto range mode Tracking

    WriteToSR785 "I2MD 0" 'Set Channel 2 input mode single-ended
    WriteToSR785 "I2GD 0" 'Set Channel 2 Input Grounding to Float
    WriteToSR785 "I2CP 1" 'Set Channel 2 Input Coupling to AC
    WriteToSR785 "I2AR 1" 'Set Channel 2 Auto range mode Tracking

    WriteToSR785 "IAOM 0" 'Set Auto offset Off
    WriteToSR785 "FAVM " & Channel & " ,2" 'Set type of averaging RMS
    WriteToSR785 "FAVT " & Channel & " ,1" 'Set Averaging type
Exponential/Continuous
    WriteToSR785 "FAVN " & Channel & " ,500" 'Set Number of averages 500

    WriteToSR785 "PDST 3" 'Set the Print/Plot/Dump destination GPIB Port
    WriteToSR785 "PCIC 0" 'Set the GPIB control mode Host

    WriteToSR785 "FLIN " & Channel & " ,3" 'Set Resolution to 800

    WriteToSR785 "KEYP 37" 'press Auto scale A

    WriteToSR785 "KEYP 17" 'press Start


    If QueriesSR785("A1RG ?") = 1 Then
        fMainForm.cmdAutorange.BackColor = &HFF00&
    Else
        fMainForm.cmdAutorange.BackColor = &H8000000F
    End If


End Sub


*******************************************************************
Module1.bas
*******************************************************************
'Copyrights - Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State
University

Public fMainForm As frmMain
Public MGPIB As Integer
Global RecContinue As Boolean
Global Channel As Integer
Sub Main()
    frmSplash.Show
    frmSplash.Refresh
    Set fMainForm = New frmMain
    Load fMainForm
    Unload frmSplash
```

```
    fMainForm.Show
    InitializeGPIB
    WriteToSR785 "OUTX 0"
    ResetSR785ToNoise
    fMainForm.UpdateDisp
MGPIB = 10
End Sub

Sub Wait(Time_in_ms As Integer)
    st = Timer
    While Timer - st < Time_in_ms / 1000
        DoEvents
    Wend
End Sub
Function ChangeUnit(number As Single) As String
    If Abs(number) < 1 Then
        number = number * 1000
        unit = " m"
    End If
    If Abs(number) < 1 Then
        number = number * 1000
        unit = " u"
    End If
    If Abs(number) < 1 Then
        number = number * 1000
        unit = " n"
    End If
    If Abs(number) < 1 Then
        number = number * 1000
        unit = " p"
    End If
    ChangeUnit = Format(number, "000.0000") & unit
End Function


***********************************************************************
frmMain.frm
***********************************************************************
'Copyrights - Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State
University


VERSION 5.00
Object = "F9043C88-F6F2-101A-A3C9-08002B2F49FB#1.2#0"; "comdlg32.ocx"
Object = "831FDD16-0C5C-11D2-A9FC-0000F8754DA1#2.0#0"; "MSCOMCTL.OCX"
Begin VB.MDIForm frmMain
   BackColor       =   &H8000000C&
   Caption         =   "Viraj2004 SR785 Data Collector Program "
   ClientHeight    =   10575
   ClientLeft      =   165
   ClientTop       =   735
   ClientWidth     =   15240
   Icon            =   "frmMain.frx":0000
   LinkTopic       =   "MDIForm1"
   StartUpPosition =   3  'Windows Default
   WindowState     =   2  'Maximized
```

```
Begin VB.Timer Timer1
   Interval        =   300
   Left            =   1560
   Top             =   3000
End
Begin VB.Timer Timer2
   Enabled         =   0   'False
   Interval        =   200
   Left            =   4440
   Top             =   3000
End
Begin MSComctlLib.Toolbar Toolbar2
   Align           =   2  'Align Bottom
   Height          =   630
   Left            =   0
   TabIndex        =   6
   Top             =   9675
   Width           =   15240
   _ExtentX        =   26882
   _ExtentY        =   1111
   ButtonWidth     =   609
   ButtonHeight    =   953
   Appearance      =   1
   _Version        =   393216
   Begin VB.CheckBox chkResetWhenComp
      Caption         =   "Reset when complete"
      Height          =   375
      Left            =   11880
      TabIndex        =   31
      Top             =   120
      Width           =   1335
   End
   Begin VB.TextBox Text9
      Appearance      =   0  'Flat
      BackColor       =   &H8000000F&
      BorderStyle     =   0  'None
      BeginProperty Font
         Name            =   "Times New Roman"
         Size            =   14.25
         Charset         =   0
         Weight          =   400
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   405
      Left            =   9840
      MousePointer    =   1  'Arrow
      TabIndex        =   30
      Text            =   "# Avgs"
      Top             =   120
      Width           =   855
   End
   Begin VB.TextBox txtNoAvg
      Alignment       =   1  'Right Justify
      BackColor       =   &H00000000&
      BeginProperty Font
```

```
      Name            =   "Arial"
      Size            =   14.25
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H00FFFF00&
   Height          =   375
   Left            =   10680
   TabIndex        =   29
   Text            =   "500"
   ToolTipText     =   "Enter No of Averages"
   Top             =   120
   Width           =   1215
End
Begin VB.TextBox txtNoOfAvg
   Alignment       =   1  'Right Justify
   BackColor       =   &H00000000&
   BeginProperty Font
      Name            =   "Arial"
      Size            =   14.25
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H000000FF&
   Height          =   375
   Left            =   13920
   Locked          =   -1  'True
   TabIndex        =   27
   Text            =   "Done  "
   ToolTipText     =   "No of Avg. Currently completed"
   Top             =   120
   Width           =   1215
End
Begin VB.TextBox Text1
   Appearance      =   0  'Flat
   BackColor       =   &H8000000F&
   BorderStyle     =   0  'None
   BeginProperty Font
      Name            =   "Times New Roman"
      Size            =   14.25
      Charset         =   0
      Weight          =   400
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   Height          =   405
   Left            =   13320
   MousePointer    =   1  'Arrow
   TabIndex        =   26
   Text            =   "Avg."
```

```
      Top              =   120
      Width            =   615
   End
   Begin VB.TextBox txtCenterF
      Alignment        =   1  'Right Justify
      BackColor        =   &H00000000&
      BeginProperty Font
         Name            =   "Arial"
         Size            =   14.25
         Charset         =   0
         Weight          =   700
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      ForeColor        =   &H0000FF00&
      Height           =   375
      Left             =   4560
      TabIndex         =   25
      Text             =   "52000"
      ToolTipText      =   "Center Frequency"
      Top              =   120
      Width            =   1215
   End
   Begin VB.TextBox Text10
      Alignment        =   2  'Center
      BackColor        =   &H00000000&
      BeginProperty Font
         Name            =   "Arial"
         Size            =   14.25
         Charset         =   0
         Weight          =   700
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      ForeColor        =   &H0000FF00&
      Height           =   375
      Left             =   9000
      MousePointer     =   1  'Arrow
      TabIndex         =   14
      Text             =   "Hz"
      Top              =   120
      Width            =   615
   End
   Begin VB.TextBox txtEndF
      Alignment        =   1  'Right Justify
      BackColor        =   &H00000000&
      BeginProperty Font
         Name            =   "Arial"
         Size            =   14.25
         Charset         =   0
         Weight          =   700
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
```

```
      ForeColor       =   &H0000FF00&
      Height          =   375
      Left            =   7800
      TabIndex        =   13
      Text            =   "102400"
      ToolTipText     =   "End Frequency"
      Top             =   120
      Width           =   1215
   End
   Begin VB.TextBox Text8
      Appearance      =   0  'Flat
      BackColor       =   &H8000000F&
      BorderStyle     =   0  'None
      BeginProperty Font
         Name            =   "Times New Roman"
         Size            =   14.25
         Charset         =   0
         Weight          =   400
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   405
      Left            =   6600
      MousePointer    =   1  'Arrow
      TabIndex        =   12
      Text            =   "End Freq. "
      Top             =   120
      Width           =   1215
   End
   Begin VB.TextBox Text7
      Alignment       =   2  'Center
      BackColor       =   &H00000000&
      BeginProperty Font
         Name            =   "Arial"
         Size            =   14.25
         Charset         =   0
         Weight          =   700
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H0000FF00&
      Height          =   375
      Left            =   5760
      MousePointer    =   1  'Arrow
      TabIndex        =   11
      Text            =   "Hz"
      Top             =   120
      Width           =   615
   End
   Begin VB.TextBox Text5
      Appearance      =   0  'Flat
      BackColor       =   &H8000000F&
      BorderStyle     =   0  'None
      BeginProperty Font
         Name            =   "Times New Roman"
```

```
      Size            =   14.25
      Charset         =   0
      Weight          =   400
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   Height          =   405
   Left            =   3120
   MousePointer    =   1  'Arrow
   TabIndex        =   10
   Text            =   "Center Freq."
   Top             =   120
   Width           =   1455
End
Begin VB.TextBox Text4
   Alignment       =   2  'Center
   BackColor       =   &H00000000&
   BeginProperty Font
      Name            =   "Arial"
      Size            =   14.25
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H0000FF00&
   Height          =   375
   Left            =   2400
   MousePointer    =   1  'Arrow
   TabIndex        =   9
   Text            =   "Hz"
   Top             =   120
   Width           =   615
End
Begin VB.TextBox Text3
   Appearance      =   0  'Flat
   BackColor       =   &H8000000F&
   BorderStyle     =   0  'None
   BeginProperty Font
      Name            =   "Times New Roman"
      Size            =   14.25
      Charset         =   0
      Weight          =   400
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   Height          =   405
   Left            =   0
   MousePointer    =   1  'Arrow
   TabIndex        =   8
   Text            =   "Start Freq."
   Top             =   120
   Width           =   1215
End
```

```
      Begin VB.TextBox txtStartF
         Alignment       =   1  'Right Justify
         BackColor       =   &H00000000&
         BeginProperty Font
            Name            =   "Arial"
            Size            =   14.25
            Charset         =   0
            Weight          =   700
            Underline       =   0   'False
            Italic          =   0   'False
            Strikethrough   =   0   'False
         EndProperty
         ForeColor       =   &H0000FF00&
         Height          =   375
         Left            =   1200
         TabIndex        =   7
         Text            =   "0"
         ToolTipText     =   "Starting Frequency"
         Top             =   120
         Width           =   1215
      End
   End
   Begin MSComctlLib.Toolbar Toolbar1
      Align           =   3  'Align Left
      Height          =   9255
      Left            =   0
      TabIndex        =   4
      Top             =   420
      Width           =   1035
      _ExtentX        =   1826
      _ExtentY        =   16325
      ButtonWidth     =   609
      ButtonHeight    =   953
      Appearance      =   1
      _Version        =   393216
      BorderStyle     =   1
      Begin VB.OptionButton optDual
         Caption         =   "Dual"
         Height          =   540
         Left            =   105
         Style           =   1  'Graphical
         TabIndex        =   36
         Top             =   3570
         Width           =   750
      End
      Begin VB.OptionButton optSingle
         Caption         =   "Single"
         Height          =   540
         Left            =   105
         Style           =   1  'Graphical
         TabIndex        =   35
         Top             =   3045
         Value           =   -1  'True
         Width           =   750
      End
      Begin VB.CommandButton cmdLiveShow
         Caption         =   "SR785"
```

```
         Height          =   810
         Left            =   0
         Picture         =   "frmMain.frx":0442
         Style           =   1  'Graphical
         TabIndex        =   34
         ToolTipText     =   "To access SR785 front panal button"
         Top             =   4560
         Width           =   975
      End
      Begin VB.CommandButton cmdRecordWhenAvg
         Caption         =   "Record 1 when Avg. complete "
         Height          =   690
         Left            =   0
         TabIndex        =   28
         ToolTipText      =    "To record spectrum after one avaragin cycle is
completed"
         Top             =   720
         Width           =   975
      End
      Begin VB.CommandButton cmdRecContinually
         Caption         =   "Record Continually"
         Height          =   690
         Left            =   0
         Style           =   1  'Graphical
         TabIndex        =   24
         ToolTipText     =   "To record spectrum end of every avaraging cycle"
         Top             =   1440
         Width           =   975
      End
      Begin VB.CommandButton cmdRec1Curve
         Caption         =   "Record One Curve"
         Height          =   690
         Left            =   0
         TabIndex        =   23
         ToolTipText     =   "To record current spectrum  "
         Top             =   0
         Width           =   975
      End
      Begin VB.CommandButton cmdSR785
         Caption         =   "SR785"
         Height          =   810
         Left            =   0
         Picture         =   "frmMain.frx":0F17
         Style           =   1  'Graphical
         TabIndex        =   5
         ToolTipText     =   "To access SR785 front panal button"
         Top             =   2160
         Width           =   975
      End
      Begin VB.Frame Frame1
         BackColor       =   &H00C00000&
         Height          =   3735
         Left            =   0
         TabIndex        =   15
         Top             =   5400
         Width           =   975
         Begin VB.CommandButton cmdReset
```

```
            Caption         =   "Full RESET"
            BeginProperty Font
               Name            =   "MS Serif"
               Size            =   8.25
               Charset         =   0
               Weight          =   400
               Underline       =   0   'False
               Italic          =   0   'False
               Strikethrough   =   0   'False
            EndProperty
            Height          =   450
            Left            =   120
            TabIndex        =   22
            ToolTipText      =   "To Reset SR785 its original configaration and
set to noise measurment"
            Top             =   3120
            Width           =   735
         End
         Begin VB.CommandButton cmdAutoScale
            Caption         =   "Auto Scale"
            BeginProperty Font
               Name            =   "MS Serif"
               Size            =   8.25
               Charset         =   0
               Weight          =   400
               Underline       =   0   'False
               Italic          =   0   'False
               Strikethrough   =   0   'False
            EndProperty
            Height          =   450
            Left            =   120
            TabIndex        =   21
            ToolTipText     =   "Auto Scale SR785 Display"
            Top             =   720
            Width           =   735
         End
         Begin VB.CommandButton cmdResetNoise
            Caption         =   "Reset to noise"
            BeginProperty Font
               Name            =   "MS Serif"
               Size            =   8.25
               Charset         =   0
               Weight          =   400
               Underline       =   0   'False
               Italic          =   0   'False
               Strikethrough   =   0   'False
            EndProperty
            Height          =   450
            Left            =   120
            TabIndex        =   20
            ToolTipText     =   "To Reset SR785 for Noise measurments"
            Top             =   2640
            Width           =   735
         End
         Begin VB.CommandButton cmdSpanDown
            Caption         =   "Span Down"
            BeginProperty Font
```

```
      Name            =   "MS Serif"
      Size            =   8.25
      Charset         =   0
      Weight          =   400
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   Height          =   450
   Left            =   120
   TabIndex        =   18
   ToolTipText     =   "Change Frequency to lower span"
   Top             =   1680
   Width           =   735
End
Begin VB.CommandButton cmdReStart
   Caption         =   "ReStart"
   BeginProperty Font
      Name            =   "MS Serif"
      Size            =   8.25
      Charset         =   0
      Weight          =   400
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   Height          =   450
   Left            =   120
   TabIndex        =   17
   ToolTipText     =   "Restart Avaraging"
   Top             =   240
   Width           =   735
End
Begin VB.CommandButton cmdSpanUp
   Caption         =   "Span    Up"
   BeginProperty Font
      Name            =   "MS Serif"
      Size            =   8.25
      Charset         =   0
      Weight          =   400
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   Height          =   450
   Left            =   120
   TabIndex        =   16
   ToolTipText     =   "Change Frequency to upper span"
   Top             =   1200
   Width           =   735
End
Begin VB.CommandButton cmdAutoRange
   Caption         =   "Auto Range"
   BeginProperty Font
      Name            =   "MS Serif"
      Size            =   8.25
      Charset         =   0
```

```
            Weight          =   400
            Underline       =   0   'False
            Italic          =   0   'False
            Strikethrough   =   0   'False
         EndProperty
         Height          =   450
         Left            =   120
         Style           =   1  'Graphical
         TabIndex        =   19
         Top             =   2160
         Width           =   735
      End
   End
End
Begin MSComDlg.CommonDialog dlgCommonDialog
   Left            =   1560
   Top             =   1800
   _ExtentX        =   847
   _ExtentY        =   847
   _Version        =   393216
End
Begin MSComctlLib.StatusBar sbStatusBar
   Align           =   2  'Align Bottom
   Height          =   270
   Left            =   0
   TabIndex        =   0
   Top             =   10305
   Width           =   15240
   _ExtentX        =   26882
   _ExtentY        =   476
   _Version        =   393216
   BeginProperty Panels 8E3867A5-8586-11D1-B16A-00C0F0283628
      NumPanels       =   3
      BeginProperty Panel1 8E3867AB-8586-11D1-B16A-00C0F0283628
         AutoSize        =   1
         Object.Width           =   21246
         Text            =   "Status"
         TextSave        =   "Status"
      EndProperty
      BeginProperty Panel2 8E3867AB-8586-11D1-B16A-00C0F0283628
         Style           =   6
         AutoSize        =   2
         TextSave        =   "4/20/2007"
      EndProperty
      BeginProperty Panel3 8E3867AB-8586-11D1-B16A-00C0F0283628
         Style           =   5
         AutoSize        =   2
         TextSave        =   "8:37 AM"
      EndProperty
   EndProperty
End
Begin MSComctlLib.ImageList imlToolbarIcons
   Left            =   2160
   Top             =   1800
   _ExtentX        =   1005
   _ExtentY        =   1005
   BackColor       =   -2147483643
```

```
ImageWidth      =   16
ImageHeight     =   16
MaskColor       =   12632256
_Version        =   393216
BeginProperty Images  2C247F25-8591-11D1-B16A-00C0F0283628}
   NumListImages   =   17
   BeginProperty ListImage1  2C247F27-8591-11D1-B16A-00C0F0283628}
      Picture         =   "frmMain.frx":19EC
      Key             =   "New"
   EndProperty
   BeginProperty ListImage2  2C247F27-8591-11D1-B16A-00C0F0283628}
      Picture         =   "frmMain.frx":1AFE
      Key             =   "Open"
   EndProperty
   BeginProperty ListImage3  2C247F27-8591-11D1-B16A-00C0F0283628}
      Picture         =   "frmMain.frx":1C10
      Key             =   "Save"
   EndProperty
   BeginProperty ListImage4  2C247F27-8591-11D1-B16A-00C0F0283628}
      Picture         =   "frmMain.frx":1D22
      Key             =   "Multi"
   EndProperty
   BeginProperty ListImage5  2C247F27-8591-11D1-B16A-00C0F0283628}
      Picture         =   "frmMain.frx":4234
      Key             =   "IV"
   EndProperty
   BeginProperty ListImage6  2C247F27-8591-11D1-B16A-00C0F0283628}
      Picture         =   "frmMain.frx":6746
      Key             =   "Setting"
   EndProperty
   BeginProperty ListImage7  2C247F27-8591-11D1-B16A-00C0F0283628}
      Picture         =   "frmMain.frx":9018
      Key             =   "Print"
   EndProperty
   BeginProperty ListImage8  2C247F27-8591-11D1-B16A-00C0F0283628}
      Picture         =   "frmMain.frx":912A
      Key             =   "Cut"
   EndProperty
   BeginProperty ListImage9  2C247F27-8591-11D1-B16A-00C0F0283628}
      Picture         =   "frmMain.frx":923C
      Key             =   "Copy"
   EndProperty
   BeginProperty ListImage10  2C247F27-8591-11D1-B16A-00C0F0283628}
      Picture         =   "frmMain.frx":934E
      Key             =   "Paste"
   EndProperty
   BeginProperty ListImage11  2C247F27-8591-11D1-B16A-00C0F0283628}
      Picture         =   "frmMain.frx":9460
      Key             =   "Bold"
   EndProperty
   BeginProperty ListImage12  2C247F27-8591-11D1-B16A-00C0F0283628}
      Picture         =   "frmMain.frx":9572
      Key             =   "Italic"
   EndProperty
   BeginProperty ListImage13  2C247F27-8591-11D1-B16A-00C0F0283628}
      Picture         =   "frmMain.frx":9684
      Key             =   "Underline"
```

```
      EndProperty
      BeginProperty ListImage14  2C247F27-8591-11D1-B16A-00C0F0283628}
         Picture         =   "frmMain.frx":9796
         Key             =   "Align Left"
      EndProperty
      BeginProperty ListImage15  2C247F27-8591-11D1-B16A-00C0F0283628}
         Picture         =   "frmMain.frx":98A8
         Key             =   "Center"
      EndProperty
      BeginProperty ListImage16  2C247F27-8591-11D1-B16A-00C0F0283628}
         Picture         =   "frmMain.frx":99BA
         Key             =   "Align Right"
      EndProperty
      BeginProperty ListImage17  2C247F27-8591-11D1-B16A-00C0F0283628}
         Picture         =   "frmMain.frx":9ACC
         Key             =   "Drawing"
      EndProperty
   EndProperty
End
Begin MSComctlLib.Toolbar tbToolBar
   Align           =   1  'Align Top
   Height          =   420
   Left            =   0
   Negotiate       =   -1  'True
   TabIndex        =   1
   Top             =   0
   Width           =   15240
   _ExtentX        =   26882
   _ExtentY        =   741
   ButtonWidth     =   609
   ButtonHeight    =   582
   Appearance      =   1
   ImageList       =   "imlToolbarIcons"
   _Version        =   393216
   BeginProperty Buttons  66833FE8-8583-11D1-B16A-00C0F0283628}
      NumButtons      =   23
      BeginProperty Button1  66833FEA-8583-11D1-B16A-00C0F0283628}
         Key             =   "New"
         Object.ToolTipText     =   "New"
         ImageKey        =   "New"
      EndProperty
      BeginProperty Button2  66833FEA-8583-11D1-B16A-00C0F0283628}
         Key             =   "Open"
         Object.ToolTipText     =   "Open"
         ImageKey        =   "Open"
      EndProperty
      BeginProperty Button3  66833FEA-8583-11D1-B16A-00C0F0283628}
         Key             =   "Save"
         Object.ToolTipText     =   "Save"
         ImageKey        =   "Save"
      EndProperty
      BeginProperty Button4  66833FEA-8583-11D1-B16A-00C0F0283628}
         Style           =   3
      EndProperty
      BeginProperty Button5  66833FEA-8583-11D1-B16A-00C0F0283628}
         Key             =   "Multi"
         Object.ToolTipText     =   "Multimeter"
```

```
   ImageKey        =   "Multi"
EndProperty
BeginProperty Button6  66833FEA-8583-11D1-B16A-00C0F0283628}
   Key             =   "IV"
   Object.ToolTipText     =   "Get IV Curve"
   ImageKey        =   "IV"
EndProperty
BeginProperty Button7  66833FEA-8583-11D1-B16A-00C0F0283628}
   Key             =   "Setting"
   Object.ToolTipText     =   "Change Advance Setting"
   ImageKey        =   "Setting"
EndProperty
BeginProperty Button8  66833FEA-8583-11D1-B16A-00C0F0283628}
   Style           =   3
EndProperty
BeginProperty Button9  66833FEA-8583-11D1-B16A-00C0F0283628}
   Key             =   "Print"
   Object.ToolTipText     =   "Print"
   ImageKey        =   "Print"
EndProperty
BeginProperty Button10  66833FEA-8583-11D1-B16A-00C0F0283628}
   Style           =   3
EndProperty
BeginProperty Button11  66833FEA-8583-11D1-B16A-00C0F0283628}
   Key             =   "Cut"
   Object.ToolTipText     =   "Cut"
   ImageKey        =   "Cut"
EndProperty
BeginProperty Button12  66833FEA-8583-11D1-B16A-00C0F0283628}
   Key             =   "Copy"
   Object.ToolTipText     =   "Copy"
   ImageKey        =   "Copy"
EndProperty
BeginProperty Button13  66833FEA-8583-11D1-B16A-00C0F0283628}
   Key             =   "Paste"
   Object.ToolTipText     =   "Paste"
   ImageKey        =   "Paste"
EndProperty
BeginProperty Button14  66833FEA-8583-11D1-B16A-00C0F0283628}
   Style           =   3
EndProperty
BeginProperty Button15  66833FEA-8583-11D1-B16A-00C0F0283628}
   Key             =   "Bold"
   Object.ToolTipText     =   "Bold"
   ImageKey        =   "Bold"
EndProperty
BeginProperty Button16  66833FEA-8583-11D1-B16A-00C0F0283628}
   Key             =   "Italic"
   Object.ToolTipText     =   "Italic"
   ImageKey        =   "Italic"
EndProperty
BeginProperty Button17  66833FEA-8583-11D1-B16A-00C0F0283628}
   Key             =   "Underline"
   Object.ToolTipText     =   "Underline"
   ImageKey        =   "Underline"
EndProperty
BeginProperty Button18  66833FEA-8583-11D1-B16A-00C0F0283628}
```

```
         Style           =   3
      EndProperty
      BeginProperty Button19  66833FEA-8583-11D1-B16A-00C0F0283628}
         Key             =   "Align Left"
         Object.ToolTipText     =   "Align Left"
         ImageKey        =   "Align Left"
         Style           =   2
      EndProperty
      BeginProperty Button20  66833FEA-8583-11D1-B16A-00C0F0283628}
         Key             =   "Center"
         Object.ToolTipText     =   "Center"
         ImageKey        =   "Center"
         Style           =   2
      EndProperty
      BeginProperty Button21  66833FEA-8583-11D1-B16A-00C0F0283628}
         Key             =   "Align Right"
         Object.ToolTipText     =   "Align Right"
         ImageKey        =   "Align Right"
         Style           =   2
      EndProperty
      BeginProperty Button22  66833FEA-8583-11D1-B16A-00C0F0283628}
         Style           =   3
      EndProperty
      BeginProperty Button23  66833FEA-8583-11D1-B16A-00C0F0283628}
         Key             =   "Drawing"
         Object.ToolTipText     =   "Drawing"
         ImageKey        =   "Drawing"
      EndProperty
   EndProperty
   Begin VB.TextBox Text11
      Appearance      =   0  'Flat
      BackColor       =   &H8000000F&
      BorderStyle     =   0  'None
      BeginProperty Font
         Name            =   "Times New Roman"
         Size            =   14.25
         Charset         =   0
         Weight          =   400
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   405
      Left            =   11160
      MousePointer    =   1  'Arrow
      TabIndex        =   33
      Text            =   "Timer"
      Top             =   0
      Width           =   735
   End
   Begin VB.TextBox txtTimer
      Alignment       =   2  'Center
      BackColor       =   &H00000000&
      BeginProperty DataFormat
         Type            =   1
         Format          =   "H:mm:ss"
         HaveTrueFalseNull=   0
```

```
         FirstDayOfWeek  =   0
         FirstWeekOfYear =   0
         LCID            =   1033
         SubFormatType   =   4
      EndProperty
      BeginProperty Font
         Name            =   "BankGothic Lt BT"
         Size            =   14.25
         Charset         =   0
         Weight          =   700
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H0000FF00&
      Height          =   375
      Left            =   11880
      TabIndex        =   32
      Text            =   "00:00:00"
      ToolTipText     =   "To Change GPIB Address of the Source Meter "
      Top             =   0
      Width           =   1935
   End
   Begin VB.TextBox Text2
      Appearance      =   0  'Flat
      BackColor       =   &H8000000F&
      BorderStyle     =   0  'None
      BeginProperty Font
         Name            =   "Times New Roman"
         Size            =   14.25
         Charset         =   0
         Weight          =   400
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   405
      Left            =   6840
      MousePointer    =   1  'Arrow
      TabIndex        =   3
      Text            =   "GPIB Address"
      Top             =   0
      Width           =   1575
   End
   Begin VB.TextBox txtGPIB
      Alignment       =   2  'Center
      BeginProperty Font
         Name            =   "Times New Roman"
         Size            =   14.25
         Charset         =   0
         Weight          =   700
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   375
      Left            =   8520
```

```
      TabIndex        =   2
      Text            =   "2"
      ToolTipText     =   "To Change GPIB Address of the Source Meter "
      Top             =   0
      Width           =   735
   End
End
Begin VB.Menu mnuFile
   Caption         =   "&File"
   Begin VB.Menu mnuFileNew
      Caption         =   "&New"
      Shortcut        =   ^N
   End
   Begin VB.Menu mnuFileOpen
      Caption         =   "&Open..."
      Shortcut        =   ^O
   End
   Begin VB.Menu mnuFileClose
      Caption         =   "&Close"
   End
   Begin VB.Menu mnuFileBar0
      Caption         =   "-"
   End
   Begin VB.Menu mnuFileSave
      Caption         =   "&Save"
   End
   Begin VB.Menu mnuFileSaveAs
      Caption         =   "Save &As..."
   End
   Begin VB.Menu mnuFileSaveAll
      Caption         =   "Save A&ll"
   End
   Begin VB.Menu mnuFileBar1
      Caption         =   "-"
   End
   Begin VB.Menu mnuFileProperties
      Caption         =   "Propert&ies"
   End
   Begin VB.Menu mnuFileBar2
      Caption         =   "-"
   End
   Begin VB.Menu mnuFilePageSetup
      Caption         =   "Page Set&up..."
   End
   Begin VB.Menu mnuFilePrintPreview
      Caption         =   "Print Pre&view"
   End
   Begin VB.Menu mnuFilePrint
      Caption         =   "&Print..."
   End
   Begin VB.Menu mnuFileBar3
      Caption         =   "-"
   End
   Begin VB.Menu mnuFileSend
      Caption         =   "Sen&d..."
   End
   Begin VB.Menu mnuFileBar4
```

```
      Caption         =   "-"
   End
   Begin VB.Menu mnuFileMRU
      Caption         =   ""
      Index           =   1
      Visible         =   0   'False
   End
   Begin VB.Menu mnuFileMRU
      Caption         =   ""
      Index           =   2
      Visible         =   0   'False
   End
   Begin VB.Menu mnuFileMRU
      Caption         =   ""
      Index           =   3
      Visible         =   0   'False
   End
   Begin VB.Menu mnuFileBar5
      Caption         =   "-"
      Visible         =   0   'False
   End
   Begin VB.Menu mnuFileExit
      Caption         =   "E&xit"
   End
End
Begin VB.Menu mnuEdit
   Caption         =   "&Edit"
   Begin VB.Menu mnuEditUndo
      Caption         =   "&Undo"
   End
   Begin VB.Menu mnuEditBar0
      Caption         =   "-"
   End
   Begin VB.Menu mnuEditCut
      Caption         =   "Cu&t"
      Shortcut        =   ^X
   End
   Begin VB.Menu mnuEditCopy
      Caption         =   "&Copy"
      Shortcut        =   ^C
   End
   Begin VB.Menu mnuEditPaste
      Caption         =   "&Paste"
      Shortcut        =   ^V
   End
   Begin VB.Menu mnuEditPasteSpecial
      Caption         =   "Paste &Special..."
   End
End
Begin VB.Menu mnuView
   Caption         =   "&View"
   Begin VB.Menu mnuViewToolbar
      Caption         =   "&Toolbar"
      Checked         =   -1  'True
   End
   Begin VB.Menu mnuViewStatusBar
      Caption         =   "Status &Bar"
```

```
      Checked         =   -1  'True
   End
   Begin VB.Menu mnuViewBar0
      Caption         =   "-"
   End
   Begin VB.Menu mnuViewRefresh
      Caption         =   "&Refresh"
   End
   Begin VB.Menu mnuViewOptions
      Caption         =   "&Options..."
   End
   Begin VB.Menu mnuViewWebBrowser
      Caption         =   "&Web Browser"
   End
End
Begin VB.Menu mnuTools
   Caption         =   "&Tools"
   Begin VB.Menu Calibrate
      Caption         =   "&MRD500 Calibration Data"
      Begin VB.Menu OpenCalFile
         Caption         =   "Open MRD500 Calibration File"
      End
      Begin VB.Menu SaveCalFile
         Caption         =   "Save As MRD500 Calibration File"
      End
   End
   Begin VB.Menu mnuToolsOptions
      Caption         =   "&Settings..."
   End
End
Begin VB.Menu mnuWindow
   Caption         =   "&Window"
   WindowList      =   -1  'True
   Begin VB.Menu mnuWindowNewWindow
      Caption         =   "&New Window"
   End
   Begin VB.Menu mnuWindowBar0
      Caption         =   "-"
   End
   Begin VB.Menu mnuWindowCascade
      Caption         =   "&Cascade"
   End
   Begin VB.Menu mnuWindowTileHorizontal
      Caption         =   "Tile &Horizontal"
   End
   Begin VB.Menu mnuWindowTileVertical
      Caption         =   "Tile &Vertical"
   End
   Begin VB.Menu mnuWindowArrangeIcons
      Caption         =   "&Arrange Icons"
   End
End
Begin VB.Menu mnuHelp
   Caption         =   "&Help"
   Begin VB.Menu mnuHelpContents
      Caption         =   "&Contents"
   End
```

```
      Begin VB.Menu mnuHelpSearchForHelpOn
         Caption         =   "&Search For Help On..."
      End
      Begin VB.Menu mnuHelpBar0
         Caption         =   "-"
      End
      Begin VB.Menu mnuHelpAbout
         Caption         =   "&About "
      End
   End
End
Attribute VB_Name = "frmMain"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Private Bpress As Boolean
Private Declare Function SendMessage Lib "user32" Alias "SendMessageA" (ByVal
hwnd As Long, ByVal wMsg As Long, ByVal wParam As Long, ByVal lParam As Any) As
Long
Const EM_UNDO = &HC7
Private Declare Function OSWinHelp% Lib "user32" Alias "WinHelpA" (ByVal hwnd&,
ByVal HelpFile$, ByVal wCommand%, dwData As Any)
Private ReSetComp As Boolean

Private Sub chkResetWhenComp_Click()
    ReSetComp = chkResetWhenComp.value
End Sub

Private Sub cmdAutorange_Click()
    WriteToSR785 "KEYP 38"
    If Channel = 2 Then
        WriteToSR785 "KEYP 46"
    End If
    If QueriesSR785("A1RG ?") = 1 Then
        cmdAutorange.BackColor = &HFF00&
    Else
        cmdAutorange.BackColor = &H8000000F
    End If
End Sub

Private Sub cmdAutoScale_Click()
    WriteToSR785 "KEYP 37"
    WriteToSR785 "KEYP 45"
End Sub

Private Sub cmdLiveShow_Click()
frmLive.Visible = True
End Sub

Private Sub cmdRec1Curve_Click()
    ActiveForm.PlotCurve
End Sub

Private Sub cmdRecContinually_Click()
    If RecContinue = False Then
        RecContinue = True
```

```
        cmdRecContinually.BackColor = &HFF&
        Timer2.Enabled = True
    Else
        RecContinue = False
        cmdRecContinually.BackColor = &H8000000F
        Timer2.Enabled = False
        Timer1.Enabled = True
    End If
End Sub

Private Sub cmdRecordWhenAvg_Click()
    RecContinue = False
    WriteToSR785 "KEYP 17"
    Timer2.Enabled = True
End Sub

Private Sub cmdReset_Click()
    nl = Chr(10) & Chr(13)
    cho = MsgBox("This will RESET SR785 to its original configaration" & nl &
"and reset parameters as suitable for Noise measurments." & nl & "Do you want
to continue ?", vbYesNo, "Change system setings")
    If cho = 6 Then
        Timer2.Enabled = False
        WriteToSR785 "*RST" 'To Reset SR785
        Wait 20000
        MsgBox "aa"
        ResetSR785ToNoise
        Timer1.Enabled = False
    End If
End Sub

Private Sub cmdResetNoise_Click()
      ResetSR785ToNoise
End Sub

Private Sub cmdReStart_Click()
    WriteToSR785 "KEYP 17"
End Sub

Private Sub cmdSpanDown_Click()
    WriteToSR785 "KEYP 61"
    fMainForm.UpdateDisp
End Sub

Private Sub cmdSpanUp_Click()
    WriteToSR785 "KEYP 53"
    fMainForm.UpdateDisp
End Sub

Private Sub cmdSR785_Click()
'    If frmSR785.Visible = False Then
        frmSR785.Visible = True
'    Else
'        frmSR785.Visible = False
'   End If
End Sub
```

```
Private Sub MDIForm_Load()
    Me.Left = GetSetting(App.Title, "Settings", "MainLeft", 1000)
    Me.Top = GetSetting(App.Title, "Settings", "MainTop", 1000)
    Me.Width = GetSetting(App.Title, "Settings", "MainWidth", 7500)
    Me.Height = GetSetting(App.Title, "Settings", "MainHeight", 6500)
    MGPIB = GetSetting(App.Title, "Settings", "GPIB", 2)
    Me.txtGPIB = MGPIB
    ReSetComp = False
    LoadNewDoc
    Channel = 0 ' for single channel
End Sub

Private Sub LoadNewDoc()
    Static lDocumentCount As Long
    Dim frmD As frmDocument
    lDocumentCount = lDocumentCount + 1
    Set frmD = New frmDocument
    frmD.Caption = "Noise Spectrum Book " & lDocumentCount
    frmD.Width = Me.Width - 1400
    frmD.Height = Me.Height - 2200
    frmD.Show
End Sub

Private Sub MDIForm_Unload(Cancel As Integer)
    If Me.WindowState <> vbMinimized Then
        SaveSetting App.Title, "Settings", "MainLeft", Me.Left
        SaveSetting App.Title, "Settings", "MainTop", Me.Top
        SaveSetting App.Title, "Settings", "MainWidth", Me.Width
        SaveSetting App.Title, "Settings", "MainHeight", Me.Height
        SaveSetting App.Title, "Settings", "GPIB", MGPIB
    End If
    ilonl Dev%, 0
    Close All
    End
End Sub

Private Sub optDual_Click()
    If optDual.value = True Then
        Channel = 2 ' for both channel
        frmDualRecord.Visible = True
        DoEvents
        ResetSR785ToNoise
    End If
End Sub

Private Sub optSingle_Click()
    If optSingle.value = True Then
        Channel = 0
        DoEvents
        ResetSR785ToNoise
    End If
End Sub

Private Sub tbToolBar_ButtonClick(ByVal Button As MSComctlLib.Button)
    On Error Resume Next
    Select Case Button.Key
        Case "New"
```

```
            LoadNewDoc
        Case "Open"
            mnuFileOpen_Click
        Case "Save"
            mnuFileSave_Click
        Case "Multi"
            Static v
            If v = 0 Then v = 1 Else v = 0
            frmSR785.Visible = v
            Button.value = v
        Case "IV"

        Case "Setting"

        Case "Print"
            mnuFilePrint_Click
        Case "Cut"
            mnuEditCut_Click
        Case "Copy"
            mnuEditCopy_Click
        Case "Paste"
            mnuEditPaste_Click
        Case "Bold"
          '  ActiveForm.rtfText.SelBold = Not ActiveForm.rtfText.SelBold
          '      Button.Value   =   IIf(ActiveForm.rtfText.SelBold,   tbrPressed,
tbrUnpressed)
        Case "Italic"
           ' ActiveForm.rtfText.SelItalic = Not ActiveForm.rtfText.SelItalic
          '     Button.Value   =   IIf(ActiveForm.rtfText.SelItalic,   tbrPressed,
tbrUnpressed)
        Case "Underline"
        '                        ActiveForm.rtfText.SelUnderline      =      Not
ActiveForm.rtfText.SelUnderline
         '     Button.Value  =  IIf(ActiveForm.rtfText.SelUnderline,  tbrPressed,
tbrUnpressed)
        Case "Align Left"
         '   ActiveForm.rtfText.SelAlignment = rtfLeft
        Case "Center"
         '   ActiveForm.rtfText.SelAlignment = rtfCenter
        Case "Align Right"
         '   ActiveForm.rtfText.SelAlignment = rtfRight
        Case "Drawing"
            'ToDo: Add 'Drawing' button code.
    End Select
End Sub

Private Sub mnuHelpAbout_Click()
    frmAbout.Show vbModal, Me
End Sub

Private Sub mnuHelpSearchForHelpOn_Click()
    Dim nRet As Integer


    'if there is no helpfile for this project display a message to the user
    'you can set the HelpFile for your application in the
    'Project Properties dialog
```

```
    If Len(App.HelpFile) = 0 Then
        MsgBox "Unable to display Help Contents. There is no Help associated
with this project.", vbInformation, Me.Caption
    Else
        On Error Resume Next
        nRet = OSWinHelp(Me.hwnd, App.HelpFile, 261, 0)
        If Err Then
            MsgBox Err.Description
        End If
    End If

End Sub

Private Sub mnuHelpContents_Click()
    Dim nRet As Integer


    'if there is no helpfile for this project display a message to the user
    'you can set the HelpFile for your application in the
    'Project Properties dialog
    If Len(App.HelpFile) = 0 Then
        MsgBox "Unable to display Help Contents. There is no Help associated
with this project.", vbInformation, Me.Caption
    Else
        On Error Resume Next
        nRet = OSWinHelp(Me.hwnd, App.HelpFile, 3, 0)
        If Err Then
            MsgBox Err.Description
        End If
    End If

End Sub


Private Sub mnuWindowArrangeIcons_Click()
    Me.Arrange vbArrangeIcons
End Sub

Private Sub mnuWindowTileVertical_Click()
    Me.Arrange vbTileVertical
End Sub

Private Sub mnuWindowTileHorizontal_Click()
    Me.Arrange vbTileHorizontal
End Sub

Private Sub mnuWindowCascade_Click()
    Me.Arrange vbCascade
End Sub

Private Sub mnuWindowNewWindow_Click()
    LoadNewDoc
End Sub

Private Sub mnuToolsOptions_Click()
    frmOptions.Show vbModal, Me
End Sub
```

```
Private Sub mnuViewOptions_Click()
    frmOptions.Show vbModal, Me
End Sub

Private Sub mnuViewRefresh_Click()
    'ToDo: Add 'mnuViewRefresh_Click' code.
    MsgBox "Add 'mnuViewRefresh_Click' code."
End Sub

Private Sub mnuViewStatusBar_Click()
    mnuViewStatusBar.Checked = Not mnuViewStatusBar.Checked
    sbStatusBar.Visible = mnuViewStatusBar.Checked
End Sub

Private Sub mnuViewToolbar_Click()
    mnuViewToolbar.Checked = Not mnuViewToolbar.Checked
    tbToolBar.Visible = mnuViewToolbar.Checked
End Sub

Private Sub mnuEditPasteSpecial_Click()
    'ToDo: Add 'mnuEditPasteSpecial_Click' code.
    MsgBox "Add 'mnuEditPasteSpecial_Click' code."
End Sub

Private Sub mnuEditPaste_Click()
    On Error Resume Next
  '  ActiveForm.rtfText.SelRTF = Clipboard.GetText

End Sub

Private Sub mnuEditCopy_Click()
    On Error Resume Next
   ' Clipboard.SetText ActiveForm.rtfText.SelRTF

End Sub

Private Sub mnuEditCut_Click()
    On Error Resume Next
   ' Clipboard.SetText ActiveForm.rtfText.SelRTF
  '  ActiveForm.rtfText.SelText = vbNullString

End Sub

Private Sub mnuEditUndo_Click()
    'ToDo: Add 'mnuEditUndo_Click' code.
    MsgBox "Add 'mnuEditUndo_Click' code."
End Sub


Private Sub mnuFileExit_Click()
    ilonl Dev%, 0
    Close All
    End
End Sub

Private Sub mnuFileSend_Click()
```

```
    'ToDo: Add 'mnuFileSend_Click' code.
    MsgBox "Add 'mnuFileSend_Click' code."
End Sub

Private Sub mnuFilePrint_Click()
    On Error Resume Next
    If ActiveForm Is Nothing Then Exit Sub

    With dlgCommonDialog
        .DialogTitle = "Print"
        .CancelError = True
        .Flags = cdlPDReturnDC + cdlPDNoPageNums
     '   If ActiveForm.rtfText.SelLength = 0 Then
     '       .Flags = .Flags + cdlPDAllPages
     '   Else
     '       .Flags = .Flags + cdlPDSelection
     '   End If
        .ShowPrinter
'        If err <> MSComDlg.cdlCancel Then
'            ActiveForm.rtfText.SelPrint .hDC
'        End If
    End With

End Sub

Private Sub mnuFilePrintPreview_Click()
    'ToDo: Add 'mnuFilePrintPreview_Click' code.
    MsgBox "Add 'mnuFilePrintPreview_Click' code."
End Sub

Private Sub mnuFilePageSetup_Click()
    On Error Resume Next
    With dlgCommonDialog
        .DialogTitle = "Page Setup"
        .CancelError = True
        .ShowPrinter
    End With

End Sub

Private Sub mnuFileProperties_Click()
    'ToDo: Add 'mnuFileProperties_Click' code.
    MsgBox "Add 'mnuFileProperties_Click' code."
End Sub

Private Sub mnuFileSaveAll_Click()
    'ToDo: Add 'mnuFileSaveAll_Click' code.
    MsgBox "Add 'mnuFileSaveAll_Click' code."
End Sub

Private Sub mnuFileSaveAs_Click()
    If ActiveForm.Caption <> "Viraj2003 Multimeter" Then
        ActiveForm.SaveAs
    Else
        frmSR785.WindowState = 1
        ActiveForm.SaveAs
        frmSR785.WindowState = 0
```

```
    End If
End Sub

Private Sub mnuFileSave_Click()
    If ActiveForm.Caption <> "Viraj2003 Multimeter" Then
        ActiveForm.Save
    Else
        frmSR785.WindowState = 1
        ActiveForm.Save
        frmSR785.WindowState = 0
    End If
    End Sub

Private Sub mnuFileClose_Click()
    'ToDo: Add 'mnuFileClose_Click' code.
    MsgBox "Add 'mnuFileClose_Click' code."
End Sub

Private Sub mnuFileOpen_Click()
    LoadNewDoc
    ActiveForm.Fopen
End Sub

Private Sub mnuFileNew_Click()
    LoadNewDoc
End Sub

Public Sub multioff()
tbToolBar.Buttons.Item(5).value = tbrUnpressed
End Sub

Private Sub Timer1_Timer()
    nAvg = QueriesSR785("NAVG ? 0")
    If nAvg > Val(txtNoAvg) And ReSetComp = True Then
        WriteToSR785 "KEYP 17"
    End If
    txtNoOfAvg.Text = nAvg
    DoEvents
End Sub

Private Sub Timer2_Timer()
    nAvg = QueriesSR785("NAVG ? 0")

    If nAvg < Val(txtNoAvg) Then
        txtNoOfAvg.Text = nAvg
        DoEvents
    Else
        Timer2.Enabled = False
        DoEvents
        ActiveForm.PlotCurve
    End If
End Sub

Private Sub txtCenterF_Validate(Cancel As Boolean)
    If txtStartF >= 0 Then
        a$ = "FCTR 0, " & txtCenterF
        WriteToSR785 a$
```

```
        UpdateDisp
    End If
End Sub

Private Sub txtEndF_Validate(Cancel As Boolean)
    If txtStartF >= 0 Then
        a$ = "FSPN 0, " & txtEndF
        WriteToSR785 a$
        UpdateDisp
    End If

End Sub

Private Sub txtGPIB_Change()
    MGPIB = txtGPIB.Text
End Sub

Private Sub txtStart_Change()
    ActiveForm.SetScale txtStart, txtStop
End Sub

Private Sub txtStop_Change()
    ActiveForm.SetScale txtStart, txtStop
End Sub
Sub UpdateDisp()
    txtStartF.Text = Format(QueriesSR785("FSTR ? 0"), "#0.0##")
    txtCenterF.Text = Format(QueriesSR785("FCTR ? 0"), "#####")
    txtEndF.Text = Format(QueriesSR785("FSPN ? 0"), "######")
End Sub

Private Sub txtNoAvg_Change()
    a$ = "FAVN 0," & txtNoAvg
    WriteToSR785 a$
End Sub

Private Sub txtStartF_Validate(Cancel As Boolean)
    If txtStartF >= 0 Then
        a$ = "FSTR 0, " & txtStartF
        WriteToSR785 a$
        UpdateDisp
    End If
End Sub
```

```
*********************************************************************
frmDocument.frm
*********************************************************************
'Copyrights - Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State
University
```

```
    Dim oBook As Excel.Workbook
    Dim oSheet As Excel.Worksheet
    Dim oChart As Excel.Chart
    Dim NoG As Long
    Dim data(1 To 1000, 1 To 2) As Single

Private Sub cmbDescription_Click()
    txtCellNo = cmbDescription.ListIndex + 1
    DescripChange
End Sub

Private Sub cmbDescription_DropDown()
    DescripChange
End Sub

Private Sub cmbDescription_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        DescripChange
    End If
End Sub

Private Sub cmbDescription_LostFocus()
    DescripChange
End Sub
Sub DescripChange()
    cmbDescription.List(txtCellNo - 1) = cmbDescription.Text
    cmbDescription.ListIndex = txtCellNo - 1
    If NoG > 0 Then
        oSheet.Cells(3, (2 * txtCellNo)).value = cmbDescription.List(txtCellNo
- 1)
    End If
End Sub

Private Sub Form_Load()
    cmbDescription.List(0) = "Noise Curve "
    cmbDescription.ListIndex = 0
    txtCellNo = cmbDescription.ListIndex + 1

    OLE1.CreateEmbed "", "excel.chart"
    Set oBook = OLE1.object
    Set oChart = oBook.Charts(1)
    Set oSheet = oBook.Worksheets(1)

    oSheet.Cells.Clear
    oChart.ChartType = xlXYScatterSmoothNoMarkers
    oChart.SetSourceData oSheet.Range("A3:B804"), xlColumns
    With oChart

        .HasTitle = True
        .ChartTitle.AutoScaleFont = False
        .ChartTitle.Font.Size = 12
        .ChartTitle.Text = "Noise Spectrum of the Sample"
        .HasLegend = True
        .Legend.Interior.ColorIndex = xlNone
        .Legend.Border.LineStyle = xlNone
        .Legend.AutoScaleFont = False
        .Legend.Font.Size = 8
```

```
        .Legend.Left = 200
        .Legend.Top = 60
        .Axes(xlCategory, xlPrimary).HasTitle = True
        .Axes(xlCategory, xlPrimary).AxisTitle.AutoScaleFont = False
        .Axes(xlCategory, xlPrimary).AxisTitle.Font.Size = 12
        .Axes(xlCategory,  xlPrimary).AxisTitle.Characters.Text  =  "Frequency
(Hz)"
        .Axes(xlValue, xlPrimary).HasTitle = True
        .Axes(xlValue,  xlPrimary).AxisTitle.Characters.Text  =  "Noise  Voltage
Density (Vrms²/Hz)"
        .Axes(xlValue, xlPrimary).AxisTitle.AutoScaleFont = False
        .Axes(xlValue, xlPrimary).AxisTitle.Font.Size = 12
    End With
    With oChart.Axes(xlValue)
        .HasMajorGridlines = False
        .HasMinorGridlines = False
        .MajorTickMark = xlInside
        .MinorTickMark = xlInside
        .TickLabelPosition = xlNextToAxis
        .TickLabels.AutoScaleFont = False
'        .TickLabels.NumberFormat = "0.00"
        .TickLabels.Font.Size = 12
        '.MinimumScale = 1
        '.MaximumScale = 1
      '  .Crosses = xlCustom
     '   .CrossesAt = .MinimumScale
        .ScaleType = xlLogarithmic
    End With
    With oChart.Axes(xlCategory)
        .HasMajorGridlines = False
        .HasMinorGridlines = False
        .MajorTickMark = xlInside
        .MinorTickMark = xlInside
        .TickLabelPosition = xlNextToAxis
        .TickLabels.AutoScaleFont = False
 '      .TickLabels.NumberFormat = "0.00"
        .TickLabels.Font.Size = 12
        '.MinimumScale = 400
        '.MaximumScale = 1200
        .ScaleType = xlLogarithmic
    End With

    oChart.PlotArea.Interior.ColorIndex = xlNone

    OLE1.Visible = True
 '  OLE1.Close  'Deactivate the OLE container
    Application.Assistant.Visible = True
    With oSheet
        .Cells(1, 1).value = "Viraj2004 Noise Spectrum Recorder Data File"
        .Cells(2, 1).value = "Vrms²/Hz vs Freq."
    End With
End Sub
Private Sub Form_Resize()
    On Error Resume Next
    OLE1.Move 100, 480, Me.ScaleWidth - 200, Me.ScaleHeight - 1100
    sldpos.Left = 120
    sldpos.Top = Me.ScaleHeight - 500
```

```
    sldpos.Width = Me.ScaleWidth - 240

End Sub
Sub Fopen()
    Dim sFile As String
    With CommonDialog
        .DialogTitle = "Open"
        .CancelError = False
        .Filter = "Excel File (*.xls)|*.xls"
        .DialogTitle = "Open Noise Spectrum File"
        .ShowOpen
        If Len(.filename) = 0 Then
            Exit Sub
        End If
        sFile = .filename
    End With
    Caption = sFile

    OLE1.CreateEmbed sFile, "excel.chart"
    Set oBook = OLE1.object
    Set oSheet = oBook.Worksheets(1)

    With oSheet
        If Left(.Cells(1, 1).value, 43) <> "Viraj2004 Noise Spectrum Recorder
Data File" Then
            MsgBox "This is not a Viraj 2003 Action spectrum Data file",
vbOKOnly + vbInformation, "Invalied file format"
            Exit Sub
        End If
        Set oChart = oBook.Charts(1)
        c = 1
        xx = .Cells(1, c).value
        While xx <> ""
            c = c + 2
            xx = .Cells(1, c).value
        Wend
    End With
    NoG = (c - 1) / 2
    For z% = 1 To NoG
        cmbDescription.List(z% - 1) = oSheet.Cells(3, 2 * z%).value
    Next z%
    cmbDescription.AddItem "Sample No " & LTrim(Str(NoG + 1))
End Sub

Sub Save()
    Dim sFile As String
    If Left$(Caption, 19) = "Noise Spectrum Book" Then
        On Error GoTo ErrHandler
        With CommonDialog
            .CancelError = True
            .Filter = "Excel File (*.xls)|*.xls"
            .DialogTitle = "Save Noise Spectrum Data File"
            .Flags = cdlOFNOverwritePrompt
            .ShowSave
            If Len(.filename) = 0 Then
                Exit Sub
            End If
```

```
            sFile = .filename
        End With
        oBook.SaveCopyAs sFile
        Caption = sFile
    Else
        sFile = Caption
        oBook.SaveCopyAs sFile
    End If
ErrHandler:
    Exit Sub
End Sub

Sub SaveAs()
    Dim sFile As String
    On Error GoTo ErrHandler
    With CommonDialog
        .CancelError = True
        .Filter = "Excel File (*.xls)|*.xls"
        .DialogTitle = "Save Noise Spectrum Data File As"
        .Flags = cdlOFNOverwritePrompt
        .ShowSave
        If Len(.filename) = 0 Then
            Exit Sub
        End If
        sFile = .filename
    End With
    oBook.SaveCopyAs sFile
    Caption = sFile

ErrHandler:
    Exit Sub
End Sub

Sub PlotCurve()
    'Dim data(1 To 810, 1 To 2) As Single
    fMainForm.Timer1.Enabled = False
    fMainForm.Timer2.Enabled = False
    fMainForm.tbToolBar.Enabled = False
    MousePointer = 11
    WriteToSR785 "KEYP 25"

    NoG = NoG + 1
    Cn% = (NoG * 2) - 1
    Rn% = 4

    With oSheet
        .Cells(3, Cn%).value = "f (Hz)"
        If NoG > 1 Then
            If   (Right(cmbDescription.List(NoG   -   2),   2)   =   "AM"   Or
Right(cmbDescription.List(NoG - 2), 2) = "PM") And Len(cmbDescription.List(NoG
- 2)) > 12 Then
                cmbDescription.List(NoG - 1) = Left(cmbDescription.List(NoG -
2), Len(cmbDescription.List(NoG - 2)) - 11) & " " & Time
            Else
                cmbDescription.List(NoG - 1) = "Noise Curve " & Trim(Str(NoG))
& "  " & Time
            End If
```

```
        Else
            cmbDescription.List(NoG - 1) = cmbDescription.List(0) & "  " & Time
        End If
        .Cells(3, Cn% + 1).value = cmbDescription.List(NoG - 1)

        cmbDescription.ListIndex = NoG - 1
        txtCellNo = cmbDescription.ListIndex + 1
        DoEvents
        n = QueriesSR785("DSPN ? 0") - 1
        StartF = QueriesSR785("FSPN ? 0") / n

        fMainForm.txtNoOfAvg.Text = "Reading..."
        fMainForm.sbStatusBar.Panels(1).Text = "Data Collecting in Progress..."
        WriteToSR785 "TONE 50,55"
        DoEvents

        WriteToSR785 "DSPY ? 0"
        a$ = ReadSR785()

        WriteToSR785 "TONE 50,55"

        fMainForm.sbStatusBar.Panels(1).Text = "Data Plotting..."

        WriteToSR785 "KEYP 17"
        fMainForm.Timer1.Enabled = True

        .Cells(Rn%, Cn%).value = StartF
        .Cells(Rn%, Cn% + 1).value = Val(Mid(a$, 17, 15))
         Rn% = Rn% + 1
       If NoG > 1 Then
            oChart.SeriesCollection.NewSeries
            oChart.SeriesCollection(NoG).XValues   =   .Range(.Cells(4,   Cn%),
.Cells(804, Cn%))
            oChart.SeriesCollection(NoG).Values  =  .Range(.Cells(4,  Cn%  +  1),
.Cells(804, Cn% + 1))
        End If
        oChart.SeriesCollection(NoG).Name   =   .Range(.Cells(3,   Cn%   +   1),
.Cells(3, Cn% + 1))
'######################################################################
        For i = 2 To n
            'data(i, 1) = i * StartF
          '  data(i, 2) = Val(Mid(a$, (16 * i) + 1, 15))

            .Cells(Rn%, Cn%).value = i * StartF
            .Cells(Rn%, Cn% + 1).value = Val(Mid(a$, (16 * i) + 1, 15))
            Rn% = Rn% + 1
            DoEvents
        Next i
        '.Range(Cells(Rn%, Cn%), Cells(Rn% + n, Cn% + 1)).value = data()
        '.Range(Cells(2, Cn%), Cells(2 + n, Cn% + 1)).value = data()

'######################################################################

    End With
    oChart.Axes(xlValue).Crosses = xlCustom
    oChart.Axes(xlValue).CrossesAt = oChart.Axes(xlValue).MinimumScale
```

```
    fMainForm.sbStatusBar.Panels(1).Text = "Done"

    If RecContinue = True Then
        fMainForm.Timer1.Enabled = False
        fMainForm.Timer2.Enabled = True
    Else
        fMainForm.Timer1.Enabled = True
    End If

    fMainForm.tbToolBar.Enabled = True
    MousePointer = 0
End Sub
Private Sub txtArea_Change()
    If txtArea = "" Then txtArea = 0
    If txtArea > 0 Then
        With oSheet
            Voc = .Cells(4, txtCellNo * 2).value
            Isc = .Cells(5, txtCellNo * 2).value
            FF = .Cells(6, txtCellNo * 2).value
            Eff = .Cells(7, txtCellNo * 2).value
            area = .Cells(10, txtCellNo * 2).value
            Eff = Eff * area / txtArea
            .Cells(7, txtCellNo * 2).value = Format(Eff, "#0.000")
            .Cells(10, txtCellNo * 2).value = Val(txtArea)
        End With
    End If
End Sub

Private Sub txtCellNo_Click()
    CommonDialog.ShowColor
    If NoG > 0 Then
        With oChart.SeriesCollection(Val(txtCellNo))
            .Border.Color = CommonDialog.Color
        End With
    End If
End Sub

Private Sub txtIntencity_Change()
    If txtIntencity = "" Then txtIntencity = 0
    If txtIntencity > 0 Then
        With oSheet
            Voc = .Cells(4, txtCellNo * 2).value
            Isc = .Cells(5, txtCellNo * 2).value
            FF = .Cells(6, txtCellNo * 2).value
            Eff = .Cells(7, txtCellNo * 2).value
            Inten = .Cells(11, txtCellNo * 2).value
            Eff = Eff * Inten / txtIntencity
            .Cells(7, txtCellNo * 2).value = Format(Eff, "#0.000")
            .Cells(11, txtCellNo * 2).value = Val(txtIntencity)
        End With
    End If
End Sub

Sub SetScale(Min As Single, Max As Single)
    oChart.Axes(xlCategory).MinimumScale = Min
    oChart.Axes(xlCategory).MaximumScale = Max
End Sub
```

```
**********************************************************************
frmSR785.frm
**********************************************************************
'Copyrights - Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State
University


Private Sub cmdAutorange_Click()
    WriteToSR785 "KEYP 38"
    If QueriesSR785("A1RG ?") = 1 Then
        lblAutoRange.Visible = True
    Else
        lblAutoRange.Visible = False
    End If
End Sub

Private Sub cmdAutoScale_Click()
    WriteToSR785 "KEYP 37"
End Sub

Private Sub cmdAvg_Click()
    i = QueriesSR785("FLIN ? 0")
    If i = 3 Then
        i = 0
    Else
        i = i + 1
    End If
    a$ = "FLIN 0, " & i
    WriteToSR785 a$
    lblFFTLines.Caption = Trim(Str(100 * 2 ^ i))
End Sub

Private Sub cmdPauseCont_Click()
    WriteToSR785 "KEYP 25"
End Sub

Private Sub cmdSpanDwn_Click()
    WriteToSR785 "KEYP 61"
    lblStartF.Caption = Format(QueriesSR785("FSTR ? 0"), "#0.0## Hz")
    lblEndF.Caption = Format(QueriesSR785("FSPN ? 0"), "##### Hz")
    fMainForm.UpdateDisp
End Sub

Private Sub cmdSpanUp_Click()
    WriteToSR785 "KEYP 53"
    lblStartF.Caption = Format(QueriesSR785("FSTR ? 0"), "#0.0## Hz")
    lblEndF.Caption = Format(QueriesSR785("FSPN ? 0"), "##### Hz")
    fMainForm.UpdateDisp
End Sub

Private Sub cmdStart_Click()
    WriteToSR785 "KEYP 17"
End Sub
```

```
Private Sub Form_Load()
    If QueriesSR785("A1RG ?") = 1 Then
        lblAutoRange.Visible = True
    Else
        lblAutoRange.Visible = False
    End If
    lblStartF.Caption = Format(QueriesSR785("FSTR ? 0"), "#0.0## Hz")
    lblEndF.Caption = Format(QueriesSR785("FSPN ? 0"), "##### Hz")
    lblFFTLines.Caption = 100 * 2 ^ QueriesSR785("FLIN ? 0")
End Sub

*********************************************************************
frmDualRecord.frm
*********************************************************************
'Copyrights - Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State
University

Private Sub cmdBrows_Click()
    On Error Resume Next
    With CommonDialog1
        .DialogTitle = "Enter File Name to store data"
        .CancelError = True
        .Filter = "All File (*.*)|*.*"
        .Flags = cdlOFNOverwritePrompt
        .ShowSave
        txtFileName.Text = .filename
    End With
End Sub

Private Sub cmdExit_Click()
    Close All
    frmDualRecord.Visible = False
End Sub

Private Sub cmdStart_Click()
    Timer1.Enabled = True
    Timer1_Timer
End Sub

Private Sub cmdStop_Click()
    Timer1.Enabled = False
End Sub

Private Sub Form_Load()
    Me.Height = 4260
    Me.Width = 5460

End Sub

Private Sub Timer1_Timer()
    Static cc As Long
    cc = cc + 1
    lblNoofCurves.Caption = Trim(Str(cc))
    MousePointer = 11
    WriteToSR785 "KEYP 25"
    tt$ = Str(Hour(Time)) + "-" + Str(Minute(Time)) + "-" + Str(Second(Time))
```

```
    fName = txtFileName.Text + "-" + tt$ + ".txt"
    Open fName For Output As #1
    Print #1, "f(Hz)", "A" & tt$, "B" & tt$

    n = QueriesSR785("DSPN ? 0") - 1
    StartF = QueriesSR785("FSPN ? 0") / n
    WriteToSR785 "TONE 50,55"
    WriteToSR785 "DSPY ? 0"
    a$ = ReadSR785()
    WriteToSR785 "DSPY ? 1"
    b$ = ReadSR785()
    WriteToSR785 "TONE 50,55"
    WriteToSR785 "KEYP 17"

    For i = 1 To n
        Print #1, i * StartF, Mid(a$, (16 * i) + 1, 15), Mid(b$, (16 * i) + 1,
15)
    Next i


    MousePointer = 0
    Close #1

End Sub

Private Sub txtTime_Change()
    Timer1.Interval = Val(txtTime.Text) * 60000
End Sub
```

B.4 x-z-θ stage controller

An inexpensive x-z-θ sample stage was constructed and tested. This stage can be used with an FT-IR to perform transmission measurements at different points on the samples at different incident angles with 80 μm special resolution in the x-z directions and 1-degree accuracy in the angle θ. The mechanical part of the stage was constructed by using stepper motors removed from old computer floppy disk drives. A software program was developed to control this stage using Visual Basic. The user interface is shown in Figure B.25.

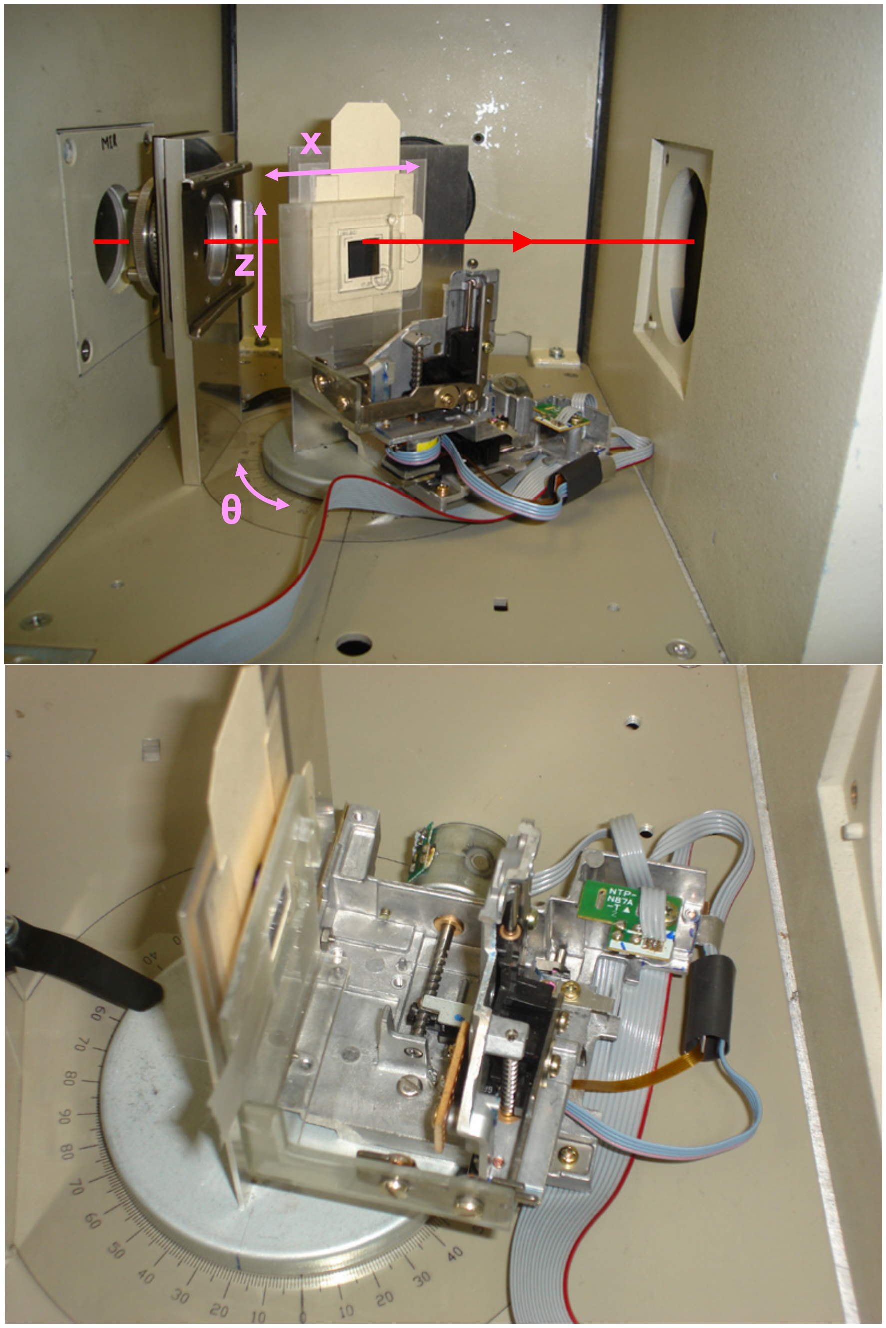


Figure B.24 Photgraphs of the x-z-θ stage (top) showing moving directions and placement in the FTIR sample compartment and (bottom) a closer view of the mechanical parts.

B.4.1 User interface

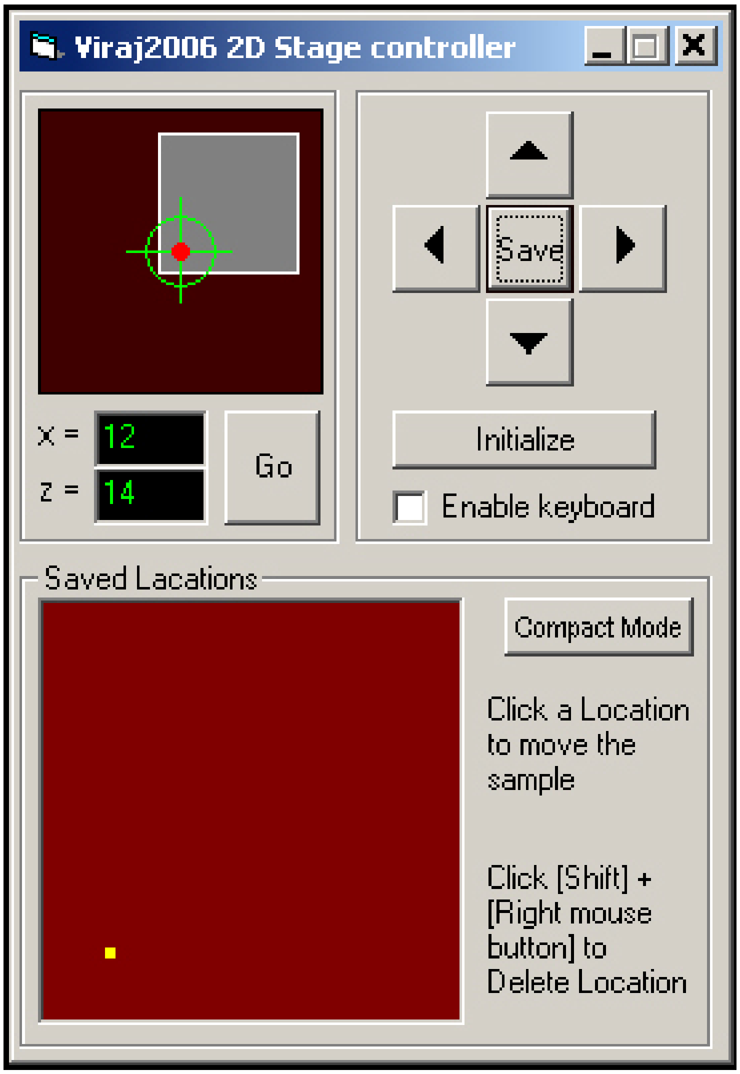


Figure B.25 The user interface of the x-z-θ stage controller software. The program allows the user to move the sample stage by clicking on-screen buttons, using keyboard arrow keys, or directly entering the new location coordinates. The user can save the locations and easily repeat the measurements on saved locations.

B.4.2 Source code

```
*********************************************************************
x-y Stage.frm
*********************************************************************
'Copyrights – Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State
University

Begin VB.Form Form1
   Caption         =   "Viraj2006 2D Stage controller"
   ClientHeight    =   5520
   ClientLeft      =   60
   ClientTop       =   345
   ClientWidth     =   3960
   LinkTopic       =   "Form1"
   MaxButton       =   0   'False
   ScaleHeight     =   5520
   ScaleWidth      =   3960
   StartUpPosition =   3  'Windows Default
   Begin VB.Frame Frame1
      Height          =   2640
      Left            =   0
      TabIndex        =   5
      Top             =   0
      Width           =   1800
      Begin VB.CommandButton cmdGo
         Caption         =   "Go"
         Height          =   645
         Left            =   1155
         TabIndex        =   11
         Top             =   1890
         Width           =   540
      End
      Begin VB.Frame Frame4
         Height          =   1600
         Left            =   100
         TabIndex        =   8
         Top             =   200
         Width           =   1600
         Begin VB.Shape Shape1
            BackColor       =   &H000000FF&
            BackStyle       =   1  'Opaque
            BorderColor     =   &H000000FF&
            Height          =   100
            Left            =   750
            Shape           =   3  'Circle
            Top             =   750
            Width           =   100
         End
         Begin VB.Shape Shape2
            BorderColor     =   &H0000FF00&
            Height          =   400
            Left            =   600
            Shape           =   3  'Circle
            Top             =   600
            Width           =   400
```

```
      End
      Begin VB.Line Line2
         BorderColor     =   &H0000FF00&
         X1              =   500
         X2              =   1100
         Y1              =   800
         Y2              =   800
      End
      Begin VB.Line Line3
         BorderColor     =   &H0000FF00&
         X1              =   795
         X2              =   795
         Y1              =   500
         Y2              =   1100
      End
      Begin VB.Shape shpSample
         BackColor       =   &H00000040&
         BackStyle       =   1  'Opaque
         BorderColor     =   &H80000005&
         FillColor       =   &H00808080&
         FillStyle       =   0  'Solid
         Height          =   800
         Left            =   800
         Top             =   0
         Width           =   800
      End
      Begin VB.Shape Shape3
         BackColor       =   &H00000040&
         BackStyle       =   1  'Opaque
         Height          =   1605
         Left            =   0
         Top             =   0
         Width           =   1605
      End
   End
   Begin VB.TextBox txtX
      BackColor       =   &H00000000&
      ForeColor       =   &H0000FF00&
      Height          =   330
      Left            =   420
      TabIndex        =   7
      Text            =   "0"
      Top             =   1890
      Width           =   645
   End
   Begin VB.TextBox txtZ
      BackColor       =   &H00000000&
      ForeColor       =   &H0000FF00&
      Height          =   330
      Left            =   420
      TabIndex        =   6
      Text            =   "0"
      Top             =   2205
      Width           =   645
   End
   Begin VB.Label Label1
      Caption         =   "x ="
```

```
      BeginProperty Font
         Name              =   "MS Sans Serif"
         Size              =   9.75
         Charset           =   0
         Weight            =   400
         Underline         =   0   'False
         Italic            =   0   'False
         Strikethrough     =   0   'False
      EndProperty
      Height          =   330
      Left            =   105
      TabIndex        =   10
      Top             =   1890
      Width           =   330
   End
   Begin VB.Label Label2
      Caption         =   "z ="
      BeginProperty Font
         Name              =   "MS Sans Serif"
         Size              =   9.75
         Charset           =   0
         Weight            =   400
         Underline         =   0   'False
         Italic            =   0   'False
         Strikethrough     =   0   'False
      EndProperty
      Height          =   330
      Left            =   105
      TabIndex        =   9
      Top             =   2205
      Width           =   330
   End
End
Begin VB.Frame Frame3
   Height          =   2640
   Left            =   1890
   TabIndex        =   4
   Top             =   0
   Width           =   2010
   Begin VB.Timer Timer2
      Enabled         =   0   'False
      Interval        =   60
      Left            =   1470
      Top             =   1470
   End
   Begin VB.CommandButton cmdDirection
      Height          =   500
      Index           =   3
      Left            =   210
      Picture         =   "x-y Stage.frx":0000
      Style           =   1  'Graphical
      TabIndex        =   20
      Top             =   735
      Width           =   500
   End
   Begin VB.CommandButton cmdDirection
      Height          =   500
```

```
      Index           =   2
      Left            =   735
      Picture         =   "x-y Stage.frx":007E
      Style           =   1  'Graphical
      TabIndex        =   19
      Top             =   1260
      Width           =   500
   End
   Begin VB.CommandButton cmdDirection
      Height          =   500
      Index           =   1
      Left            =   1260
      Picture         =   "x-y Stage.frx":00F7
      Style           =   1  'Graphical
      TabIndex        =   18
      Top             =   735
      Width           =   500
   End
   Begin VB.CommandButton cmdInitialize
      Caption         =   "Initialize"
      Height          =   330
      Left            =   210
      TabIndex        =   12
      Top             =   1890
      Width           =   1485
   End
   Begin VB.CheckBox chkKeyBoard
      Caption         =   "Enable keyboard"
      Height          =   255
      Left            =   210
      TabIndex        =   2
      Top             =   2310
      Width           =   1650
   End
   Begin VB.Timer Timer1
      Enabled         =   0   'False
      Interval        =   50
      Left            =   105
      Top             =   210
   End
   Begin VB.CommandButton cmdDirection
      Height          =   500
      Index           =   0
      Left            =   735
      Picture         =   "x-y Stage.frx":0175
      Style           =   1  'Graphical
      TabIndex        =   0
      Top             =   210
      Width           =   500
   End
   Begin VB.CommandButton cmdSave
      Caption         =   "Save"
      Height          =   500
      Left            =   735
      Style           =   1  'Graphical
      TabIndex        =   1
      ToolTipText     =   "Click to Save Current Location"
```

```
         Top             =   735
         Width           =   500
      End
   End
   Begin VB.Frame Frame2
      Caption         =   "Saved Lacations"
      Height          =   2745
      Left            =   0
      TabIndex        =   3
      Top             =   2730
      Width           =   3900
      Begin VB.CommandButton cmdCompact
         Caption         =   "Compact Mode"
         BeginProperty Font
            Name            =   "Arial Narrow"
            Size            =   8.25
            Charset         =   0
            Weight          =   400
            Underline       =   0   'False
            Italic          =   0   'False
            Strikethrough   =   0   'False
         EndProperty
         Height          =   330
         Left            =   2730
         TabIndex        =   16
         Top             =   210
         Width           =   1065
      End
      Begin VB.PictureBox pic1
         BackColor       =   &H00000080&
         ForeColor       =   &H0000FF00&
         Height          =   2400
         Left            =   105
         ScaleHeight     =   80
         ScaleMode       =   0  'User
         ScaleWidth      =   80
         TabIndex        =   14
         Top             =   210
         Width           =   2400
         Begin VB.Label lblPos
            Appearance      =   0  'Flat
            BackColor       =   &H0000FFFF&
            ForeColor       =   &H80000008&
            Height          =   60
            Index           =   0
            Left            =   0
            MouseIcon       =   "x-y Stage.frx":01C0
            MousePointer    =   99  'Custom
            TabIndex        =   17
            Top             =   2340
            Width           =   60
         End
      End
      Begin VB.Label Label4
         Caption          =   "Click [Shift] + [Right mouse button] to Delete
Location"
         Height          =   855
```

```
         Left            =   2625
         TabIndex        =   15
         Top             =   1680
         Width           =   1170
      End
      Begin VB.Label Label3
         Caption         =   "Click a Location to move the sample"
         Height          =   750
         Left            =   2625
         TabIndex        =   13
         Top             =   735
         Width           =   1170
      End
   End
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Public BPress As Boolean
Dim CPx As Integer
Dim CPz As Integer
Dim n As Integer
Dim DKey As Integer
Dim Location(1 To 100, 1 To 2) As Integer

Private Sub chkKeyBoard_Click()
    If chkKeyBoard.Value = 1 Then
        Timer1.Enabled = True
    Else
        Timer1.Enabled = False
    End If
End Sub

Private Sub cmdCompact_Click()
    Form1.Height = 3090
End Sub

Private Sub cmdDirection_MouseDown(Index As Integer, Button As Integer, Shift
As Integer, X As Single, Y As Single)
    DKey = Index
    Timer2.Enabled = True
End Sub

Private Sub cmdDirection_MouseUp(Index As Integer, Button As Integer, Shift As
Integer, X As Single, Y As Single)
    Timer2.Enabled = False
End Sub

Private Sub cmdGo_Click()
    GoPos Int(txtX.Text), Int(txtZ.Text)
End Sub

Private Sub cmdInitialize_Click()
    Initialize
End Sub
```

```
Private Sub lblPos_Click(Index As Integer)
    NewX% = Location(Index, 1)
    NewZ% = Location(Index, 2)
    GoPos NewX%, NewZ%
End Sub

Private  Sub  lblpos_MouseDown(Index  As  Integer,  Button  As  Integer,  Shift  As
Integer, X As Single, Y As Single)
    If Button = 2 And Shift = 1 Then
        Unload lblPos(Index)
        Location(Index, 1) = -1
        Location(Index, 2) = -1

    End If
End Sub

Private Sub cmdSave_Click()
    For i = 1 To n
        If Location(i, 1) = CPx And Location(i, 2) = CPz Then
            MsgBox "You already Saved This Location", vbExclamation
            Exit Sub
        End If
    Next i
    n = n + 1
    Form1.Height = 5925
    Load lblPos(n)
    lblPos(n).Top = 80 - CPz
    lblPos(n).Left = CPx
    lblPos(n).Visible = True
    Location(n, 1) = CPx
    Location(n, 2) = CPz
    lblPos(n).ToolTipText = "Location " & n & "  (" & CPx & "," & CPz & ")"
End Sub

Private Sub cmdUp_MouseDown(Button As Integer, Shift As Integer, X As Single, Y
As Single)
    BPress = True
    While BPress = True
       shpSample.Top = shpSample.Top - 10
       DoEvents
        n = 1
        While n < 25000
            n = n + 1
        Wend
    Wend

End Sub

Private Sub cmdUp_MouseUp(Button As Integer, Shift As Integer, X As Single, Y
As Single)
    BPress = False
End Sub
Private Sub cmdDown_MouseDown(Button As Integer, Shift As Integer, X As Single,
Y As Single)
```

```
    BPress = True
    While BPress = True
       shpSample.Top = shpSample.Top + 10
       DoEvents
        n = 1
        While n < 25000
            n = n + 1
        Wend
    Wend

End Sub

Private Sub cmdDown_MouseUp(Button As Integer, Shift As Integer, X As Single, Y
As Single)
    BPress = False

End Sub

Private Sub Command2_Click()
For ii = 1 To 10
    For i = 1 To 70
        Out &H378, 2
        Out &H378, 3
        Wait 5
        DoEvents
    Next i
    For i = 1 To 70
        Out &H378, 0
        Out &H378, 1
        Wait 5
        DoEvents
    Next i
Next ii
End Sub

Private Sub Form_Load()
'    Form1.Height = 3090
    Form1.Show
    n = 0
    Initialize
End Sub

Private Sub pic1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y
As Single)
    If Button = 1 Then
        GoPos CInt(X), CInt(80 - Y)
    End If
End Sub

Private Sub pic1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
pic1.ToolTipText = CInt(X) & "," & CInt(80 - Y)

End Sub

Private Sub Timer1_Timer()
```

```
    If GetKeyState(vbKeyDown) And CPz < 80 And KeyDown Then
        Out &H378, 10
        Out &H378, 11
        Form1.shpSample.Top = Form1.shpSample.Top + 10
        CPz = CPz + 1
        txtZ.Text = CPz
    End If
    If GetKeyState(vbKeyUp) And CPz > 0 And KeyDown Then
        Out &H378, 8
        Out &H378, 9
        Form1.shpSample.Top = Form1.shpSample.Top - 10
        CPz = CPz - 1
        txtZ.Text = CPz
    End If
    If GetKeyState(vbKeyLeft) And CPx < 80 And KeyDown Then
        Out &H378, 6
        Out &H378, 7
        Form1.shpSample.Left = Form1.shpSample.Left - 10
        CPx = CPx + 1
        txtX.Text = CPx
    End If
    If GetKeyState(vbKeyRight) And CPx > 0 And KeyDown Then
        Out &H378, 4
        Out &H378, 5
        Form1.shpSample.Left = Form1.shpSample.Left + 10
        CPx = CPx - 1
        txtX.Text = CPx
    End If
End Sub
Sub Initialize()
    ' *********************************
    ' To find z motor track zero
    ' *********************************
    Out &H378, 10
    If Inp(&H379) = 56 Then
        For i = 1 To 15
            Out &H378, 8
            Out &H378, 9
            Wait 5
            DoEvents
        Next i
    End If
    i = 1
    While i < 90 And Inp(&H379) = 120
        Out &H378, 10
        Out &H378, 11
        Wait 5
        DoEvents
        i = i + 1
    Wend
    If i = 90 Then
        MsgBox "Initialization failed.  Please check all the cables and power
supply and press [Initialize] button.", vbCritical, "Error... Initialization
Failed"
        Exit Sub
    Else
        For i = 1 To 80
```

```
            Out &H378, 8
            Out &H378, 9
            Wait 5
            DoEvents
        Next i

        Form1.shpSample.Top = 0
        CPz = 0
        txtX.Text = 0
    End If

    ' ********************************
    ' To find x motor track zero
    ' ********************************
    Out &H378, 6
    If Inp(&H379) = 56 Then
        For i = 1 To 15
            Out &H378, 4
            Out &H378, 5
            Wait 5
        Next i
    End If
    i = 1
    While i < 90 And Inp(&H379) = 120
        Out &H378, 6
        Out &H378, 7
        Wait 5
        i = i + 1
    Wend
    If i = 90 Then
        MsgBox "Initialization failed.  Please check all the cables and power
supply and press [Initialize] button.", vbCritical, "Error... Initialization
Failed"
        Exit Sub
    Else
        For i = 1 To 80
            Out &H378, 4
            Out &H378, 5
            Wait 5
        Next i
        Form1.shpSample.Left = 800
        CPx = 0
        txtZ.Text = 0
    End If

End Sub

Private Sub Timer2_Timer()
    If DKey = 2 And CPz < 80 Then
        Out &H378, 10
        Out &H378, 11
        Form1.shpSample.Top = Form1.shpSample.Top + 10
        CPz = CPz + 1
        txtZ.Text = CPz
    End If
    If DKey = 0 And CPz > 0 Then
        Out &H378, 8
```

```
        Out &H378, 9
        Form1.shpSample.Top = Form1.shpSample.Top - 10
        CPz = CPz - 1
        txtZ.Text = CPz
    End If
    If DKey = 3 And CPx < 80 Then
        Out &H378, 6
        Out &H378, 7
        Form1.shpSample.Left = Form1.shpSample.Left - 10
        CPx = CPx + 1
        txtX.Text = CPx
    End If
    If DKey = 1 And CPx > 0 Then
        Out &H378, 4
        Out &H378, 5
        Form1.shpSample.Left = Form1.shpSample.Left + 10
        CPx = CPx - 1
        txtX.Text = CPx
    End If

End Sub

Sub GoPos(X As Integer, z As Integer)
    Dx = X - CPx
    Dz = z - CPz
    If Dx > 0 Then
        For i = 1 To Dx
            Out &H378, 6
            Out &H378, 7
            Form1.shpSample.Left = Form1.shpSample.Left - 10
            CPx = CPx + 1
            If CPx > 80 Then Exit Sub
            txtX.Text = CPx
            DoEvents
            Wait 5
        Next i
    End If

    If Dx < 0 Then
        For i = 1 To Abs(Dx)
            Out &H378, 4
            Out &H378, 5
            Form1.shpSample.Left = Form1.shpSample.Left + 10
            CPx = CPx - 1
            If CPx < 0 Then Exit Sub
            txtX.Text = CPx
            DoEvents
            Wait 5
        Next i
    End If

    If Dz > 0 Then
        For i = 1 To Dz
            Out &H378, 10
            Out &H378, 11
            Form1.shpSample.Top = Form1.shpSample.Top + 10
            CPz = CPz + 1
```

```
            If CPz > 80 Then Exit Sub
            txtZ.Text = CPz
            DoEvents
            Wait 5
        Next i
    End If

    If Dz < 0 Then
        For i = 1 To Abs(Dz)
            Out &H378, 8
            Out &H378, 9
            Form1.shpSample.Top = Form1.shpSample.Top - 10
            CPz = CPz - 1
            If CPz < 0 Then Exit Sub
            txtZ.Text = CPz
            DoEvents
            Wait 5
        Next i

    End If



End Sub


**********************************************************************
module1.bas
**********************************************************************
'Copyrights – Viraj Jayaweera Piyankarage, Optoelectronics Lab of Georgia State
University




Public Declare Function GetKeyState Lib "user32" _
    (ByVal nVirtKey As Long) As Integer
Public Const KeyDown As Long = -128
Sub Wait(Time_in_ms As Integer)
    st = Timer
    While Timer - st < Time_in_ms / 1000
        DoEvents
    Wend
End Sub
```

Appendix C

Microsoft Excel macro programs development for data file manipulation

C.1 I-V-T data import and Arrhenius data extracting

The following two macro programs helps the user to import and plot multiple data files generated from the I-V-T measurement program (under the option "Save As ASCII text file") and extract Arrhenius data (See Appendix A). Figure C.1 shows the customized Excel menu items so that user can easily access the macro programs.

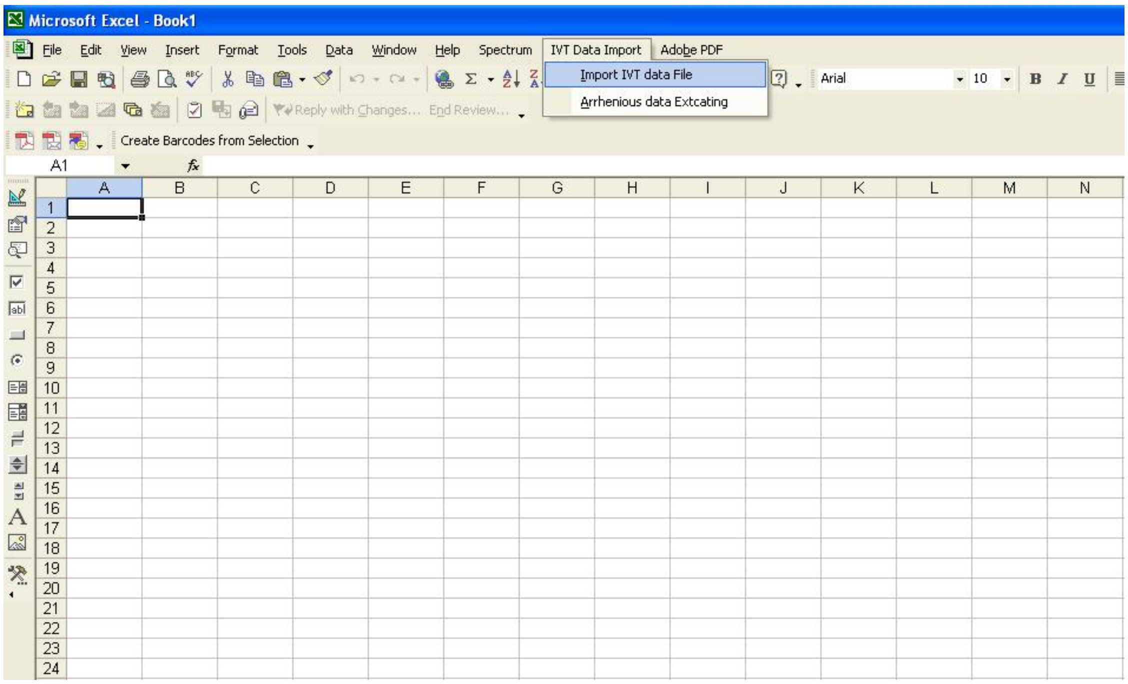


Figure C.1 Inserted custom menu showing the items "Import IVT Data" and "Arrhenius data extracting". "Import IVT Data" allows the user to import a large number of data files at once and automatically plot the data. The second item allows the user to extract Arrhenius data (See Appendix A) and save it in sheet 2 of the Excel work book.

C.1.1 Excel macro source code

```
Sub OpenIVT()
  ' Macro written by  Viraj Jayaweera Piyankarage 1/29/2006
  ' To open IVT Ascii files on excel workbook


  Dim fnames As Variant
  On Error GoTo ErrHandler
  nl = Chr(10) & Chr(13)

  fnames = Application.GetOpenFilename("txt Files (*.txt), *.txt", , , , MultiSelect:=True)
  temp = fnames(1)
  For i = 1 To UBound(fnames) - 1
    fnames(i) = fnames(i + 1)
  Next i
  fnames(UBound(fnames)) = temp

  Counter = 1
  c = ActiveCell.Column
  Workbooks.Add
  Range("A1").Select
  NewBook = ActiveWorkbook.Name

  While Counter <= UBound(fnames)

    Workbooks.OpenText Filename:= _
    fnames(Counter), Origin:=932, _
    StartRow:=1, DataType:=xlDelimited, TextQualifier:=xlDoubleQuote, _
    ConsecutiveDelimiter:=False, Tab:=True, Semicolon:=False, Comma:=True, _
    Space:=False, Other:=False, FieldInfo:=Array(Array(1, 1), Array(2, 1), Array( _
    3, 1)), TrailingMinusNumbers:=True
    Nname = ActiveWorkbook.Name
    Columns("A:B").Select
    Selection.Copy
    Windows(NewBook).Activate
    ActiveSheet.Paste
    c = ActiveCell.Column
    c = c + 2

' ***************convert to Absoloute value******************
    ActiveCell.SpecialCells(xlLastCell).Select
    r = ActiveCell.Row
    Range(Cells(20, c), Cells(20, c)).Select
    ActiveCell.FormulaR1C1 = "=ABS(RC[-1])"
```

```
        Selection.AutoFill Destination:=Range(Cells(20, c), Cells(r, c))
        Range(Cells(20, c), Cells(r, c)).Select
        Selection.Copy
        Range(Cells(20, c - 1), Cells(20, c - 1)).Select
        Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
            :=False, Transpose:=False
        Range(Cells(20, c), Cells(20, c)).Select
        Range(Selection, Selection.End(xlDown)).Select
        Application.CutCopyMode = False
        Selection.ClearContents
' *********************************************

        Range(Cells(1, c), Cells(1, c)).Select
        Windows(Nname).Activate
        ActiveWorkbook.Close
        Counter = Counter + 1
    Wend
  Windows(NewBook).Activate
  ActiveCell.SpecialCells(xlLastCell).Select
  Selection.End(xlToLeft).Select
  r = ActiveCell.Row


  Range("A20:B20").Select
    Range(Selection, Selection.End(xlDown)).Select
    Charts.Add
    ActiveChart.ChartType = xlXYScatterSmoothNoMarkers
    rr = "A20:B" & Trim(r)
    ActiveChart.SetSourceData Source:=Sheets("Sheet1").Range(rr), PlotBy _
        :=xlColumns
    ActiveChart.Location Where:=xlLocationAsNewSheet
    With ActiveChart
        .HasTitle = False
        .Axes(xlCategory, xlPrimary).HasTitle = True
        .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "Voltage (V)"
        .Axes(xlValue, xlPrimary).HasTitle = True
        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Current (A)"
        .SeriesCollection(1).Name = "=Sheet1!R18C2"

    End With
    Sheets("Chart1").Select
    ActiveChart.PlotArea.Select
    For Counter = 2 To UBound(fnames)
        ActiveChart.SeriesCollection.NewSeries
        cc = (Counter * 2) - 1
        xr = "=Sheet1!R20C" & Trim(cc) & ":R" & Trim(r) & "C" & Trim(cc)
```

```
      yr = "=Sheet1!R20C" & Trim(cc + 1) & ":R" & Trim(r) & "C" & Trim(cc + 1)
      nr = "=Sheet1!R18C" & Trim(cc + 1)
      ActiveChart.SeriesCollection(Counter).XValues = xr
      ActiveChart.SeriesCollection(Counter).Values = yr
      ActiveChart.SeriesCollection(Counter).Name = nr
   Next Counter
ErrHandler:

End Sub


Sub ArrheniusCalculation()
'
' ArrheniusCalculation Macro
' Macro written 2/11/2006 by Viraj Jayaweera Piyankarage

'
   Sheets("Sheet2").Select
   Range("A1").Select
   i = 1
   If ActiveCell.Value <> "Arrhenius data extract from Sheet 1 IVT data" Then
      ActiveCell.FormulaR1C1 = "Arrhenius data extract from Sheet 1 IVT data"
      Range("A3").Select
      ActiveCell.FormulaR1C1 = "Temparature (K)"
      Range("A4").Select

   Else

      While Val(Range(Cells(4, i), Cells(4, i)).Value) <> 0
         i = i + 1
      Wend
      i = i - 1
      Range(Cells(4, i), Cells(4, i)).Select
   End If
   Sheets("Sheet1").Select

   c1 = ActiveCell.Column
   r1 = ActiveCell.Row
   V1 = ActiveCell.Value

   I1 = Range(Cells(r1, c1 + 1), Cells(r1, c1 + 1)).Value
   T1 = Range(Cells(18, c1 + 1), Cells(18, c1 + 1)).Value

   Sheets("Sheet2").Select
   Range(Cells(3, i + 1), Cells(3, i + 1)).Value = "Bias V " & V1
   c2 = ActiveCell.Column
```

```
    r2 = ActiveCell.Row

    If c2 = 1 Then
        Range(Cells(r2, c2), Cells(r2, c2)).Value = T1
        Range(Cells(r2, c2 + 1), Cells(r2, c2 + 1)).Value = I1
    ElseIf Range(Cells(r2, 1), Cells(r2, 1)).Value = T1 Then
        Range(Cells(r2, c2 + 1), Cells(r2, c2 + 1)).Value = I1
    End If

    Sheets("Sheet1").Select

    Oldc1 = c1
    Do
        Cells.Find(What:=ActiveCell.Value, After:=ActiveCell, LookIn:=xlValues, LookAt _
            :=xlWhole, SearchOrder:=xlByColumns, SearchDirection:=xlNext, MatchCase:= _
            False, SearchFormat:=False).Activate

        c1 = ActiveCell.Column
        r1 = ActiveCell.Row
        If r1 > 19 Then
            I1 = Range(Cells(r1, c1 + 1), Cells(r1, c1 + 1)).Value
            T1 = Range(Cells(18, c1 + 1), Cells(18, c1 + 1)).Value
            Sheets("Sheet2").Select
            r2 = r2 + 1

            If c2 = 1 Then
                Range(Cells(r2, c2), Cells(r2, c2)).Value = T1
                Range(Cells(r2, c2 + 1), Cells(r2, c2 + 1)).Value = I1
            ElseIf Range(Cells(r2, 1), Cells(r2, 1)).Value = T1 Then
                Range(Cells(r2, c2 + 1), Cells(r2, c2 + 1)).Value = I1
            End If

            Sheets("Sheet1").Select
        End If
    Loop While c1 > Oldc1


End Sub
```

C.2 FT-IR Spectrum data import and responsivity calculation

The following two macro programs helps user to import and plot multiple data files generated form I-V-T measurement program (under the option save as ASCII text file) and extract Arrhenius data (See appendix A). Figure C.2 shows the customized Excel menu items so that user can easily access macro programs.

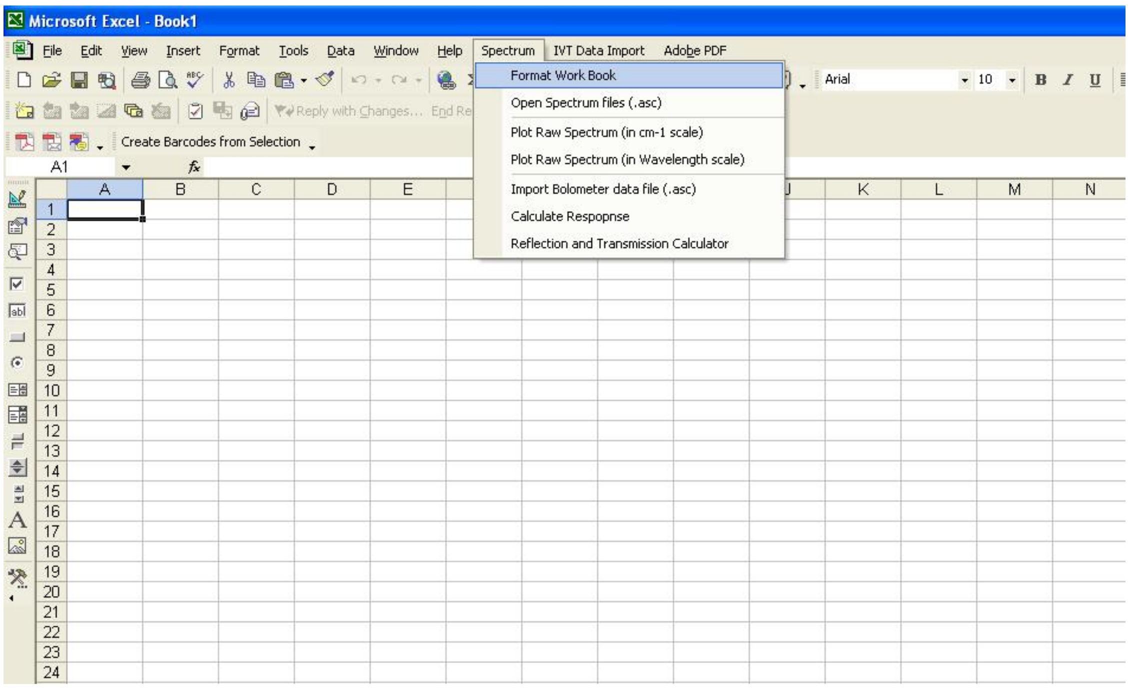


Figure C.2 Custom menu item "Spectrum" inserted in to the Excel menu allows user to: format workbook, import multiple spectrum data files, plot data, import bolometer data file, calculate responsivity. The last option can be used to import multiple data files and calculate reflection/transmission with a selected background curve.

C.2.1 Excel macro source code

```
'*******************************************************************
Sub FormatWorkSheet()
'*******************************************************************
' FormatWorkSheet Macro
' Macro recorded 7/8/2005 by Viraj Jayaweera Piyankarage
'
'
    Sheets.Add
    Sheets(1).Select
    Sheets(1).Name = "Raw data"
    Sheets(2).Select
    Sheets(2).Name = "Response data"
    Sheets(3).Select
    Sheets(3).Name = "NF"
    Sheets(4).Select
    Sheets(4).Name = "Bolometer"

    Sheets("Raw data").Select
    ActiveCell.FormulaR1C1 = "Wavenumber (cm-1)"
    With ActiveCell.Characters(Start:=1, Length:=14).Font
        .Name = "Arial"
        .FontStyle = "Regular"
        .Size = 10
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .ColorIndex = xlAutomatic
    End With
    With ActiveCell.Characters(Start:=15, Length:=2).Font
        .Name = "Arial"
        .FontStyle = "Regular"
        .Size = 10
        .Strikethrough = False
        .Superscript = True
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .ColorIndex = xlAutomatic
    End With
```

```
With ActiveCell.Characters(Start:=17, Length:=1).Font
    .Name = "Arial"
    .FontStyle = "Regular"
    .Size = 10
    .Strikethrough = False
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ColorIndex = xlAutomatic
End With
Columns("A:A").Select
Columns("A:A").EntireColumn.AutoFit
Range("A1:A2").Select
With Selection.Interior
    .ColorIndex = 38
    .Pattern = xlSolid
End With
Range("B1").Select
ActiveCell.FormulaR1C1 = "Wavelength (um)"
Columns("B:B").Select
Columns("B:B").EntireColumn.AutoFit
Range("B1:B2").Select
With Selection.Interior
    .ColorIndex = 40
    .Pattern = xlSolid
End With
Range("A1:B2").Select
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeTop)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
```

```
With Selection.Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlInsideVertical)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
Range("A3").Select


Sheets("Bolometer").Select
ActiveCell.FormulaR1C1 = "Wavenumber (cm-1)"
Range("A1").Select
ActiveCell.FormulaR1C1 = "Wavenumber (cm-1)"
With ActiveCell.Characters(Start:=1, Length:=14).Font
    .Name = "Arial"
    .FontStyle = "Regular"
    .Size = 10
    .Strikethrough = False
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ColorIndex = xlAutomatic
End With
With ActiveCell.Characters(Start:=15, Length:=2).Font
    .Name = "Arial"
    .FontStyle = "Regular"
    .Size = 10
    .Strikethrough = False
    .Superscript = True
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ColorIndex = xlAutomatic
End With
With ActiveCell.Characters(Start:=17, Length:=1).Font
    .Name = "Arial"
    .FontStyle = "Regular"
    .Size = 10
    .Strikethrough = False
```

```
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ColorIndex = xlAutomatic
End With
Columns("B:B").Select
Columns("A:A").EntireColumn.AutoFit
Range("A1").Select
With Selection.Interior
    .ColorIndex = 38
    .Pattern = xlSolid
End With
Range("B1").Select
ActiveCell.FormulaR1C1 = "Wavelength"
Columns("B:B").Select
Columns("B:B").EntireColumn.AutoFit
ActiveCell.FormulaR1C1 = "Wavelength (um)"
Range("B1").Select
Columns("B:B").EntireColumn.AutoFit
With Selection.Interior
    .ColorIndex = 40
    .Pattern = xlSolid
End With
Range("C1").Select
Selection.Interior.ColorIndex = 4
ActiveCell.FormulaR1C1 = "Bolometer Raw data"
Range("D1").Select
ActiveCell.FormulaR1C1 = "NDF filter T%"
Selection.Interior.ColorIndex = 37
Range("E1").Select
ActiveCell.FormulaR1C1 = "Si Filter T%"
Selection.Interior.ColorIndex = 45

Range("F1").Select
ActiveCell.FormulaR1C1 = "Bolometer Corected Energy"
Selection.Interior.ColorIndex = 4

Range("D1").Select
Selection.Interior.ColorIndex = 37
Range("E1").Select
Selection.Interior.ColorIndex = 45

Columns("C:F").EntireColumn.AutoFit
```

```
Range("A1:F1").Select
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
   .LineStyle = xlContinuous
   .Weight = xlThin
   .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeTop)
   .LineStyle = xlContinuous
   .Weight = xlThin
   .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeBottom)
   .LineStyle = xlContinuous
   .Weight = xlThin
   .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeRight)
   .LineStyle = xlContinuous
   .Weight = xlThin
   .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlInsideVertical)
   .LineStyle = xlContinuous
   .Weight = xlThin
   .ColorIndex = xlAutomatic
End With
Range("A2").Select

Sheets("NF").Select
Range("A1").Select
Range("A1").Value = "Rd= Dinamic Resistance of the sample"
Range("A2").Value = "Rl= Load resistance"
Range("A3").Value = "S= Sensitivity"
Range("A4").Value = "GF= gain factor (1/(1-Cos(Q))"
Range("A5").Value = "GA=Preamp Gain (200=1)"
Range("A6").Value = "Req= equlent resistor"
Range("A7").Value = ""
Range("A8").Value = "NF=S * GF/Req"
Range("A10").Select
ActiveCell.FormulaR1C1 = "Bias (V)"
Range("B10").Select
ActiveCell.FormulaR1C1 = "Rd (ohm)"
Range("C10").Select
ActiveCell.FormulaR1C1 = "Rl (ohm)"
```

```
Range("D10").Select
ActiveCell.FormulaR1C1 = "S (V/W)"
Range("E10").Select
ActiveCell.FormulaR1C1 = "GF"
Range("F10").Select
ActiveCell.FormulaR1C1 = "GA"
Range("G10").Select
ActiveCell.FormulaR1C1 = "Req (ohm)"
Range("H10").Select
ActiveCell.FormulaR1C1 = "NF"
Range("A10:H10").Select
With Selection
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlBottom
    .WrapText = False
    .Orientation = 0
    .AddIndent = False
    .IndentLevel = 0
    .ShrinkToFit = False
    .ReadingOrder = xlContext
    .MergeCells = False
End With
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeTop)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlInsideVertical)
    .LineStyle = xlContinuous
```

```
        .Weight = xlThin
        .ColorIndex = xlAutomatic
    End With
    With Selection.Interior
        .ColorIndex = 35
        .Pattern = xlSolid
    End With
    Range("A11:G20").Select
    With Selection.Interior
        .ColorIndex = 40
        .Pattern = xlSolid
    End With
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
    With Selection.Borders(xlEdgeLeft)
        .LineStyle = xlContinuous
        .Weight = xlThin
        .ColorIndex = xlAutomatic
    End With
    With Selection.Borders(xlEdgeTop)
        .LineStyle = xlContinuous
        .Weight = xlThin
        .ColorIndex = xlAutomatic
    End With
    With Selection.Borders(xlEdgeBottom)
        .LineStyle = xlContinuous
        .Weight = xlThin
        .ColorIndex = xlAutomatic
    End With
    With Selection.Borders(xlEdgeRight)
        .LineStyle = xlContinuous
        .Weight = xlThin
        .ColorIndex = xlAutomatic
    End With
    With Selection.Borders(xlInsideVertical)
        .LineStyle = xlContinuous
        .Weight = xlThin
        .ColorIndex = xlAutomatic
    End With
    With Selection.Borders(xlInsideHorizontal)
        .LineStyle = xlContinuous
        .Weight = xlThin
        .ColorIndex = xlAutomatic
    End With
    Range("H11:H20").Select
    With Selection.Interior
```

```
    .ColorIndex = 34
    .Pattern = xlSolid
End With
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeTop)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlInsideHorizontal)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
Range("A10:H10").Select
Selection.Font.Bold = True
Columns("G:G").ColumnWidth = 8.86

Range("D11").Select
ActiveCell.FormulaR1C1 = "300000"
Range("E11").Select
ActiveCell.FormulaR1C1 = "3.4142"
Range("G11").Select
ActiveCell.FormulaR1C1 = "=RC[-4]*RC[-5]/(RC[-4]+RC[-5])"
Range("H11").Select
ActiveCell.FormulaR1C1 = "=(RC[-4]*RC[-3])/(RC[-1]*RC[-2])"
Range("G11:H11").Select
Selection.AutoFill Destination:=Range("G11:H20"), Type:=xlFillDefault
Range("G11:H20").Select
Range("D11:F11").Select
```

```
    Selection.AutoFill Destination:=Range("D11:F20"), Type:=xlFillDefault
    Range("D11:F20").Select
    Range("F22").Select

    Range("A11").Select
    Sheets("Raw data").Select

End Sub



'*********************************************************************
Sub OpenSpectrum()
'*********************************************************************

    ' OpenSpectrum Macro
    ' Macro written 7/8/2005 by Viraj Jayaweera Piyankarage
    ' To open spectrum Ascii files on excel workbook


    Dim fnames As Variant
    On Error GoTo ErrHandler
    nl = Chr(10) & Chr(13)

    If Sheets(1).Name <> "Raw data" Then
        cho = MsgBox("This Work book not formated to Spectrum Response Calculations." & nl &
"Would you like to convert this?", vbYesNo, "Work Book Format!")
        If cho = 6 Then
            FormatWorkSheet
            Sheets("Raw Data").Select
        End If
    End If

    If Range("A3").Value = "" Then
        Range("A1").Select
        Selection.Interior.ColorIndex = 38
        ActiveCell.FormulaR1C1 = "Wavenumber (cm-1)"
        ActiveCell.Characters(Start:=15, Length:=2).Font.Superscript = True
        Range("B1").Select
        Selection.Interior.ColorIndex = 36
        ActiveCell.FormulaR1C1 = "Wavelength (um)"
        Range("c2").Select

    Else
        Range("B3").Select
        n = 3
```

```
    While ActiveCell.FormulaR1C1 <> ""
        Range(Cells(3, n), Cells(3, n)).Select
        n = n + 1
    Wend
End If

fnames = Application.GetOpenFilename("Ascii Files (*.asc), *.asc", , , , MultiSelect:=True)
temp = fnames(1)
For i = 1 To UBound(fnames) - 1
    fnames(i) = fnames(i + 1)
Next i
fnames(UBound(fnames)) = temp

Counter = 1
c = ActiveCell.Column

While Counter <= UBound(fnames)
    Open fnames(Counter) For Input As #1
    r = 3
    For i = 1 To 9
        Input #1, a$
    Next i
    Range(Cells(1, c), Cells(1, c)).Select
    Selection.Interior.ColorIndex = 35
    Range(Cells(1, c), Cells(1, c)).Value = "(" & Chr$(62 + c) & ") " & a$
    Range(Cells(2, c), Cells(2, c)).Value = "(" & Chr$(94 + c) & ") "
    For i = 1 To 75
        Input #1, a$
    Next i
    If c = 3 Then
        While Not EOF(1)
          Range(Cells(r, 1), Cells(r, 1)).Value = Val(Left(a$, 12))
          Range(Cells(r, c), Cells(r, c)).Value = Val(Mid(a$, 12, Len(a$)))
          Input #1, a$
          r = r + 1
        Wend
        Range("B3").FormulaR1C1 = "=10000/RC[-1]"
        Range("B3").Select
        Selection.AutoFill Destination:=Range(Cells(3, 2), Cells(r - 2, 2))

    Else
        While Not EOF(1)
            Range(Cells(r, c), Cells(r, c)).Value = Val(Mid(a$, 12, Len(a$)))
            Input #1, a$
            r = r + 1
        Wend
```

```
        End If
        Close #1
        'Range(Cells(1, c), Cells(1, c)).EntireColumn.AutoFit
        c = c + 1
        Counter = Counter + 1
    Wend
MsgBox "Done"
ErrHandler:

End Sub

'*********************************************************************
Sub OpenBolometer()
'*********************************************************************

    ' Macro written 7/8/2005 by Viraj Jayaweera Piyankarage
    ' To open Bolometer data Ascii files on excel workbook



    Dim fnames As Variant
  '  On Error GoTo ErrHandler
    nl = Chr(10) & Chr(13)

    If Sheets(1).Name <> "Raw data" Then
        cho = MsgBox("This Work book not formated to Spectrum Response Calculations." & nl &
"Would you like to convert this?", vbYesNo, "Work Book Format!")
        If cho = 6 Then
            FormatWorkSheet
        End If
    End If
    Sheets("Bolometer").Select

    If Range("A2").Value = "" Then
        Range("A1").Select
        Selection.Interior.ColorIndex = 38
        ActiveCell.FormulaR1C1 = "Wavenumber (cm-1)"
        ActiveCell.Characters(Start:=15, Length:=2).Font.Superscript = True
        Range("B1").Select
        Selection.Interior.ColorIndex = 36
        ActiveCell.FormulaR1C1 = "Wavelength (um)"
        Range("c2").Select

    Else
        Range("A2").Select
    End If
```

```
   fname = Application.GetOpenFilename("Ascii Files (*.asc), *.asc", , , , MultiSelect:=False)
   Close All
   Open fname For Input As #1
   r = 2
   For i = 1 To 9
      Input #1, a$
   Next i
   Range(Cells(1, 3), Cells(1, 3)).Select
   Selection.Interior.ColorIndex = 35
   Range(Cells(1, 3), Cells(1, 3)).Value = "Bolometer " & a$

   For i = 1 To 75
      Input #1, a$
   Next i

   While Not EOF(1)
     Range(Cells(r, 1), Cells(r, 1)).Value = Val(Left(a$, 12))
     Range(Cells(r, 3), Cells(r, 3)).Value = Val(Right(a$, 8))
     Input #1, a$
     r = r + 1
   Wend

   Range("B2").FormulaR1C1 = "=10000/RC[-1]"
   Range("B2").Select
   Selection.AutoFill Destination:=Range(Cells(2, 2), Cells(r - 1, 2))

   Close #1
   Range(Cells(1, 3), Cells(1, 3)).EntireColumn.AutoFit

ErrHandler:

End Sub

'*******************************************************************
Sub PlotRawSpectrum()
'*******************************************************************

   Range("A2").Select
'    Sheets("Raw data").Select
   Range(Selection, Selection.End(xlToRight)).Select
   Range(Selection, Selection.End(xlDown)).Select
   Charts.Add
   ActiveChart.ApplyCustomType ChartType:=xlUserDefined, TypeName:= _
      "Raw Spectrum cm-1"
   ActiveChart.SetSourceData Source:=Sheets("Raw data").Range("A1:I8502"), _
      PlotBy:=xlColumns
```

```
    ActiveChart.SeriesCollection(1).Delete
    ActiveChart.Location Where:=xlLocationAsObject, Name:="Raw data"
    With ActiveChart
        .HasTitle = True
        .ChartTitle.Characters.Text = "Raw spectrum of the sample   "
        .Axes(xlCategory, xlPrimary).HasTitle = True
        .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = _
        "Wavenumber (cm-1)"
        .Axes(xlValue, xlPrimary).HasTitle = True
        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Energy"
    End With

End Sub


Sub ResponseCal()

    Sheets("Raw data").Select
    Rows("1:2").Select
    Selection.Copy
    Sheets("Response data").Select
    Range("A1").Select
    ActiveSheet.Paste
    Range("C3").Select
    Application.CutCopyMode = False
    ActiveCell.FormulaR1C1 = "='Raw data'!RC*NF!R11C8/Bolometer!R[-1]C6"
    Range("C4").Select
    Sheets("Raw data").Select
    Range("A3:B3").Select
    Range(Selection, Selection.End(xlDown)).Select
    Selection.Copy
    Sheets("Response data").Select
    Range("A3").Select
    ActiveSheet.Paste
    Range("A3").Select

End Sub
```

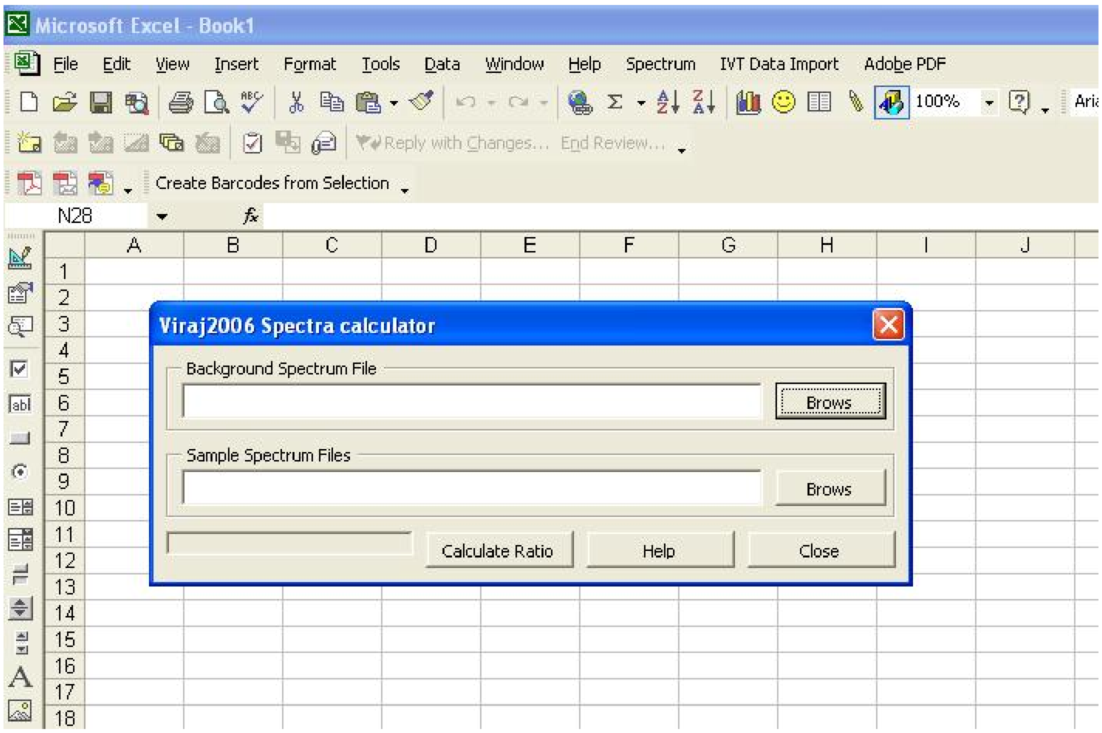


Figure C.3 The user interface of the spectral calculator macro program. The "Browse" button can be used to open Multiple sample spectra files and one background spectra file. Pressing the "Calculate Ratio" button will start copying sample data into the Sheet 1, write the calculated data into the Sheet 2 of the Excel work book, and plot the final graph.

```
Public fNameBack As Variant
Public fNameSamp As Variant
Public BData As Variant
Public NoData As Long

Private Sub cmdBrowsBack_Click()
   fNameBack = Application.GetOpenFilename("Ascii Files (*.asc), *.asc", , , ,
MultiSelect:=False)
   txtBack.Text = fNameBack
   Open fNameBack For Input As #1
   For i% = 1 To 86
      Input #1, X
   Next i%
   NoData = Val(X)
   ReDim BData(1 To NoData, 1 To 2)
   While Not EOF(1) And X <> "#DATA"
      Input #1, X
   Wend
   For i% = 1 To NoData
      Input #1, X, Y
      BData(i%, 1) = X
      If Y = 0 Then Y = 0.000001
      BData(i%, 2) = Y
   Next i%
   Close #1
End Sub

Private Sub cmdBrowsSamp_Click()
   fNameSamp = Application.GetOpenFilename("Ascii Files (*.asc), *.asc", , , ,
MultiSelect:=True)
   temp = fNameSamp(1)
   For i% = 1 To UBound(fNameSamp) - 1
      fNameSamp(i%) = fNameSamp(i% + 1)
   Next i%
   fNameSamp(UBound(fNameSamp)) = temp

   i% = 1
   While i% <= UBound(fNameSamp)
      txtSamp.Text = txtSamp.Text & fNameSamp(i%) & ","
      i% = i% + 1
   Wend
End Sub

Private Sub cmdCalc_Click()
   pgb1.Value = 1
```

```
UserForm1.MousePointer = fmMousePointerHourGlass
If Range("A2").Value = "" And fNameBack <> "" Then
   Range("A1").Select
   Selection.Interior.ColorIndex = 38
   ActiveCell.FormulaR1C1 = "Wavenumber (cm-1)"
   ActiveCell.Characters(Start:=15, Length:=2).Font.Superscript = True
   Range("B1").Select
   Selection.Interior.ColorIndex = 36
   ActiveCell.FormulaR1C1 = "Wavelength (um)"
   Range("c2").Select
   For n = 1 To NoData
      Range(Cells(n + 1, 1), Cells(n + 1, 1)).Value = BData(n, 1)
   Next n
   r = n + 1
   Range("B2").FormulaR1C1 = "=10000/RC[-1]"
   Range("B2").Select
   Selection.AutoFill Destination:=Range(Cells(2, 2), Cells(r - 1, 2))
   Range(Cells(1, 1), Cells(1, 2)).EntireColumn.AutoFit
End If
c = 1
While Range(Cells(2, c), Cells(2, c)).Value <> ""
   c = c + 1
Wend
If txtSamp.Value <> "" Then
   nf% = 1
   While nf% <= UBound(fNameSamp)
      Open fNameSamp(nf%) For Input As #1
      r = 2
      For i% = 1 To 9
         Input #1, X
      Next i%
      Range(Cells(1, c), Cells(1, c)).Select
      Selection.Interior.ColorIndex = 35
      Range(Cells(1, c), Cells(1, c)).Value = "(" & Chr$(62 + c) & ") " & X
      While Not EOF(1) And X <> "#DATA"
         Input #1, X
      Wend
      For i% = 1 To NoData
        Input #1, X, Y
        Range(Cells(r, c), Cells(r, c)).Value = Y * 100 / BData(i%, 2)
        r = r + 1
      Next i%
      pgb1.Value = nf% * 100 / UBound(fNameSamp)
      nf% = nf% + 1
      c = c + 1
      Close #1
```

```
        Wend
        c = c + 1
    End If
    UserForm1.MousePointer = fmMousePointerDefault
End Sub

Private Sub cmdClose_Click()
    UserForm1.Hide
End Sub
```